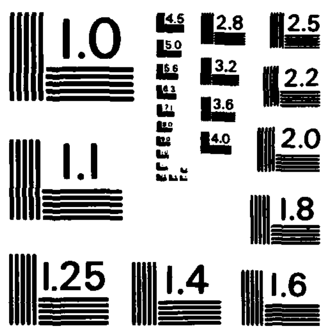
AD-A162 553 SPACE-FREQUENCY SAMPLING CRITERIA FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT(U) OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F Y FOK AUG 85 1/3

UNCLASSIFIED ESL-714190-11 N00014-82-K-0037 F/G 20/14 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(13)

OSU

The Ohio State University

AD-A162 553

SPACE-FREQUENCY SAMPLING CRITERIA
FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT

By

Fredric Y. S. Fok

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report No. 714190-11
Contract No. N00014-82-K-0037
August 1985

Department of the Navy
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

AD-A162553

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO.	2.	3. Recipient's Accession No
4. Title and Subtitle SPACE-FREQUENCY SAMPLING CRITERIA FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT			5. Report Date August 1985
			6.
7. Author(s) Frederic Y.S. Fok			8. Performing Organization Rept. No 714190-11
9. Performing Organization Name and Address Ohio State University ElectroScience Laboratory 1314 Kinnear Road Columbus, Ohio 43212			10. Project/Task/Work Unit No
			11. Contract(C) or Grant(G) No (C)N00014-82-K-0037 (G)
12. Sponsoring Organization Name and Address Department of the Navy Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217			13. Type of Report & Period Covered Technical
			14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

This investigation concerns the sampling criteria in the wave number space for generating the spatial impulse response of a finite target. The impulse response of a finite target is important for target identification and imaging. The other purpose of this report is in the management of large amounts of data for potential application in the presentation of scattered field data and construction of such images. For clarity, monostatic impulse responses of up to two dimensions are considered.

A proper choice of canonical confinement for the target in space can greatly reduce the number of samples required to sufficiently characterize the target's spatial impulse response. One dimensional impulse responses generated using two basic types of confinement: 1) isotropic and 2) parallelepiped, are compared. The two approaches show competitive reconstructed results. Though a sampling lattice may be more efficient in the sense of a reduced number of measurement points, it may be less effective when digital processing is involved. Specifically, the time consuming interpolation step is required to put data presented in other types of sampling lattice into the cubic type.

Two dimensional impulse responses reconstructed from cubic sampled data are compared with those using Mensa et al.'s method. Two dimensional impulse responses obtained also indicate good potentials for image reconstruction via the spatial impulse response.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) Unclassified	21. No of Pages 192
	20. Security Class (This Page) Unclassified	22. Price

(See ANSI–Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4–77) (Formerly NTIS–35) Department of Commerce

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONTINUED)

CHAPTER I

INTRODUCTION

Discrete data sampling at the Nyquist rate (or better) for the reconstruction of continuous waveforms is well known. However, the application of the sampling theorem [1,2] is often limited to one dimensional problems. This report will explore an application of the N dimensional sampling theorem to inverse scattering. More specifically, the application is on the reconstruction of the spatial impulse response of a finite object using different sampling criteria. The different aspects of the N dimensional sampling theorem are investigated to produce efficient sampling criteria in the wave number space such that the spatial impulse response of a finite object is sufficiently characterized. In addition, the impulse response concept is extended to create possibly an image of the object.

The impulse response [3,4] is important both in target identification and imaging. The impulse response concept, as most people understand, is a far field one dimensional time response concept. This time response waveform when applied in scattering can be plotted on a distance abscissa after the factor due to the speed of the wave is

accounted for. If one can obtain the frequency responses of a finite object for all aspect angles over the 4π solid angle, the inverse Fourier transform of the total frequency response into the spatial domain forms an image which is called the spatial impulse response. i.e., the impulse response of a finite object at $\bar{x}$ is

$$f(\bar{x}) \overset{\Delta}{=} \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} F(\bar{k}) e^{j(\bar{k}\cdot\bar{x})} d\bar{k} \tag{1-1}$$

where $F(\bar{k})$ = the far field frequency response of a finite object at $\bar{k}$ in the space-frequency domain

$$\bar{x} = (x_1, x_2, x_3)$$
$$\bar{k} = (k_1, k_2, k_3)$$
$$d\bar{k} = dk_1\, dk_2\, dk_3$$

The three dimensional function $f(\bar{x})$ for all $\bar{x}$'s in the x-space constitutes an image called the spatial impulse response.

The spatial impluse response can be divided into two types - the monostatic and the bistatic. The monostatic spatial impulse response uses monostatic frequency responses of all aspect angles. The bistatic spatial impulse response uses the bistatic frequency responses of all receiving aspect angles. The spatial, two dimensional, or one dimensional impulse responses discussed hereafter will all be referring to the monostatic case unless otherwise stated. The data used are also monostatic data. However, the theory to be discussed is also applicable to bistatic cases. Furthermore, only two dimensional frequency data are

used in the discussion, so this report will emphasize the two dimensional impulse response only.

In the introduction, the theme and purpose of this report are defined. A brief description on each of the following chapters is also included. In Chapter II, the impulse response, sampling theorem and their relationship are discussed. The basics of the one dimensional impulse response concept and the one dimensional sampling theorem are first reviewed. Using the idea of the settling time, one can relate the impulse response to the sampling theorem. Then the one dimensional impulse response concept is extended to the three dimensional space. Lewis and Bojarski's [5,6] work in the area is also briefly contrasted. A decision rule based on Petersen and Middleton's [7] definition of sampling efficiency is introduced so that one can decide on the more efficient sampling grid. There is also a discussion on Petersen and Middleton's N dimensional sampling theorem and its different forms which depend on the different types of sampling techniques. Then Mensa et al.'s [8] polar transformation and single frequency approach to two dimensional Fourier transform is rederived. Their approach presents a new perspective to the Nyquist sampling criteria.

Some practical aspects of the theory in Chapter II are considered in Chapter III. Even though the discussion is in one dimension, it is also applicable to two and three dimensional analyses. The real signal requirement on Fourier transform is rederived. This helps to clarify the problem of non-zero imaginary parts during data processing. Using the sampling theorem interpolation scheme, the frequency bandwidth's

relationship to target size and spatial resolution is considered. A common problem that may be easily overlooked in data processing is the periodicity of the Fourier series representation to solve the Fourier integral. This is also restated in the last part of Chapter III.

In order to build some confidence in the different forms of Petersen and Middleton's sampling criteria, results of interpolating one dimensional impulse responses are presented in Chapter IV. The interpolation scheme used is the same as in the sampling theorem, except the infinite summation is a finite summation. The interpolated results using one dimensional sampled data appear first in the chapter to preview the interpolation via two dimensional sampled data. Later, an example is chosen to compare the efficiency of different sampling grids. Two dimensional impulse responses computed using discrete two dimensional Fourier transform are compared with results using Mensa et al.'s approach to Fourier transformation in Chapter V. The potential of using the spatial impulse response for target imaging is explored in the last part of the report.

CHAPTER II

THEORY

Some of the basic concepts on impulse response and one dimensional sampling theorem are individually reviewed. Then the two concepts are combined and extended to three dimensional space. The different aspects of N dimensional sampling theorem are introduced. Finally, Mensa et al.'s approach to two dimensional Fourier transform and sampling is also discussed.

If an impulsive electric field is incident on a target, the normalized far zone time domain scattering will be the impulse response of the target at that particular aspect angle and polarization. The approach to be used herein to obtain the impulse response is similar to deconvolution. First, the scattered field (a complex function of frequency) is divided by the spectrum of the incident wave. This result is then inverse Fourier transformed to produce the desired impulse response.

Let $f(t)$ be the input signal in time

$F(\omega)$ be the input frequency spectrum

$h(t)$ be the impulse response of the target

$H(\omega)$ be the frequency response of the target

$c(t)$ be the output signal in time

$C(\omega)$ be the output frequency spectrum

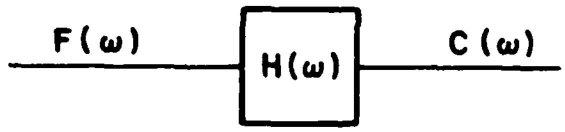


Figure 2-1. Block diagram depicting system analogy

By the convolution theorem of Fourier transform theory [2]

$$C(\omega) = F(\omega)H(\omega) \tag{2-1}$$

$$\Leftrightarrow \quad H(\omega) = \frac{C(\omega)}{F(\omega)} \tag{2-2}$$

Fourier transformed,

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{C(\omega)}{F(\omega)} e^{j\omega t} d\omega \tag{2-3}$$

According to Kennaugh and Moffatt [3], the impulse response will decay exponentially for large values of $(t - \frac{r}{c})$

where t -time

r -distance between observation point and the origin

c -speed of light

Therefore, one can define a settling time when the impulse response has its amplitude embedded in the noise level. For all practical purposes, the settling time will be the end of the impulse response. Thus, the impulse response is a time limited signal. From the sampling theorem [1,2]: "A time limited signal can be reproduced from its discrete frequency values, if it is sampled over the complete frequency domain using the Nyquist rate." As a consequence, the impulse response may be reproduced by measuring its frequency spectrum at the Nyquist sampling rate (f_s). (i.e., $f_s < 1/2T$; where the signal is limited in time to $\pm T$)

Or, if

$$f(t) = 0 \qquad |t| > T$$

then

$$F(\omega) = \sum_{n=-\infty}^{\infty} F\left(\frac{n\pi}{T}\right) \frac{\sin(\omega T - n\pi)}{\omega T - n\pi} \tag{2-4}$$

where the samples are taken at $\omega = n\pi/T$, but n is taken from $-\infty$ to ∞.

Now, let's consider the concept of marching in time. An impulsive magnetic field incident upon a solid conducting body, sets up current $\bar{J}$ on the surface. As a result, $\bar{J} = \hat{n} \times \bar{H}^T$ will start generating a scattering field in all directions. After the wave passes over the target, the current created decays exponentially. The scattered field behaves similarly. At one aspect angle, the time, where the onset of the scattered waveform is observed, corresponding to the time required for the wave to reach the initial scattering point on the target and return to the radar, is designated as the initial time T_i. The time (T_f) before the final exponential decay occurrence defines the end of the target.

Let x be the length of the object along the line of sight at one aspect angle

Δt be the time difference between the initial time and the final time at that aspect angle

c be the speed of light

$$\Delta t = T_f - T_i$$

then,

$$2x = c\Delta t$$

$$x = \frac{c(T_f - T_i)}{2} \qquad (2\text{-}5)$$

Any physical object has finite dimensions. If it is located in a rectangular coordinate system ($\bar{x}$: x_1, x_2, x_3), then it can be said to have limited dimensions in the $\bar{x}$ coordinates. Namely, the object is confined by:

$$x_1^a < x_1 < x_1^b$$

$$x_2^a < x_2 < x_2^b$$

$$x_3^a < x_3 < x_3^b$$

where x_1^a, x_2^a, x_3^a, x_1^b, x_2^b, x_3^b are some real constants.

The confinement of an object in space is equivalent in saying its spatial impulse response is time limited, as the impulse response has a settling time for every aspect angle. Using the above argument, if the impulse response measurement is obtained at all aspect angles, then an image of the object may be generated from the data.

The approach taken here is a little different from Lewis-Bojarski's work [5,6]. They use physical optics approximation to arrive at the formulation for imaging. In another words, they do not use the information on the shadowed side. The object is illuminated in every direction. Information on the 4π solid angle is required, even though measurement on the complete solid frequency sphere is difficult to be implemented. With the help of the N dimensional sampling theorem, which is described later, the implementation becomes more practical. If one requires information over a finite frequency range, then the infinite number of samples over the 4π solid angle is converted to a finite number of samples. The N dimensional sampling theorem defines a sufficient sampling criterion to characterize space limited or wave number limited signal. Thus, the response at any point may be interpolated from the sampled data using the reconstruction scheme defined by the sampling theorem.

To employ the sampling theorem, the signal must be limited in time or frequency. In this case, the object is limited in three dimensions. The transformed space will be the wave number space. Ideally, the frequency response or the k-space response can be reproduced by sampling in the k-space discretely but over the infinite space. To obtain the spatial impulse response, a three dimensional inverse Fourier transform is performed on the k-space response.

While there is only one sampling theorem, there are different forms of the sampling that satisfy the sampling theorem. In one dimension, the different forms depend on how the time limited signal or the frequency limited signal is assumed to repeat itself. In three dimensions, the forms depend on how the target is placed in the reference plane (x_1, x_2, x_3), and how the target's images are repeated in the three dimensional space. Theoretically, there are infinitely different forms of the sampling, as there is an infinite number of different target shapes, sizes and orientations. One would prefer a general sampling scheme that is applicable to all, or at least most, objects with any orientation. This is where the canonical containment cell fits in. These canonical containment cells are usually of simple geometries so that a wide variety of targets can be confined by their boundaries. Examples of these simple geometries are sphere, parallelepiped, ellipsoid, finite cylinder. Thus the forms of the sampling depend on the choice of the canonical confinement units and how the unit's images are repeated in the three dimensional space.

Two simple canonical units to be treated in this report are the sphere and parallelepiped. A decision rule between these two types of units will be discussed. First, the concept of sampling efficiency will be defined. The following efficiency formula is a modified version of the original formula defined by Petersen and Middleton [7].

$$\eta_s = \frac{\text{CONSERVATIVE ESTIMATE OF THE VOLUME OF THE OBJECT}}{\text{VOLUME OF THE SMALLEST SPHERE ENCLOSING THE OBJECT}} \qquad (2\text{-}6)$$

= Efficiency for isotropic sampling

$$\eta_p = \frac{\text{CONSERVATIVE ESTIMATE OF THE VOLUME OF THE OBJECT}}{\text{VOLUME OF THE SMALLEST PARALLELEPIPED ENCLOSING THE OBJECT}}$$

= Efficiency for parallelepipedic sampling (2-7)

Rule:

$\eta_s > \eta_p$ use spherical confinement

(2-8)

$\eta_s < \eta_p$ use parallelepiped confinement

The N dimensional sampling theorem obtained by Petersen and Middleton [7] is: "A function $F(\bar{k})$ whose inverse Fourier transform $f(\bar{x})$ vanishes over all but a finite portion in x-space can be everywhere reproduced from its sampled values taken over a lattice of points

$\{l_1\bar{u}_1 + l_2\bar{u}_2 + \ldots + l_n\bar{u}_n\}$; $l_1, l_2, \ldots, l_n = 0, \pm 1, \pm 2, \ldots$ provided that the vectors $\{\bar{u}_j\}$ are small enough to ensure non-overlapping of the x-space signal $f(\bar{x})$ with its images on a periodic lattice defined by the vectors $\{\bar{v}_i\}$, which $\bar{v}_i.\bar{u}_j = 2\pi\delta_{ij}$."

Let's consider the non-overlapping condition in the N dimensional sampling theorem. The requirement is 'non-overlapping' of the object cell $f(\bar{x})$ with its images on a periodic lattice. Thus, the periodic lattice is not uniquely defined. Any one of Figure 2-2, 2-3, or 2-4 has a valid two dimensional periodicity. Their respective periodicity is defined by their respective $\{\bar{v}_1, \bar{v}_2\}$. This non-uniqueness provides flexibility on the choice of the confinement cell.

However, an efficient sampling lattice may be defined. Pete,sen and Middleton [7]: "An efficient sampling lattice is one which uses a minimum number of sampling points to achieve an exact reproduction of a space limited function." In another words, the closest packing of the object cell and its images without overlapping will be efficient. $\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}$ will be changed if the images are rotated around the object cell; hence, the sampling lattice is still not fully defined (Figures 2-2 and 2-4). Nevertheless, any set of $\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}$ defined by the above criterion will have the same efficiency, or the same number of sampling points.

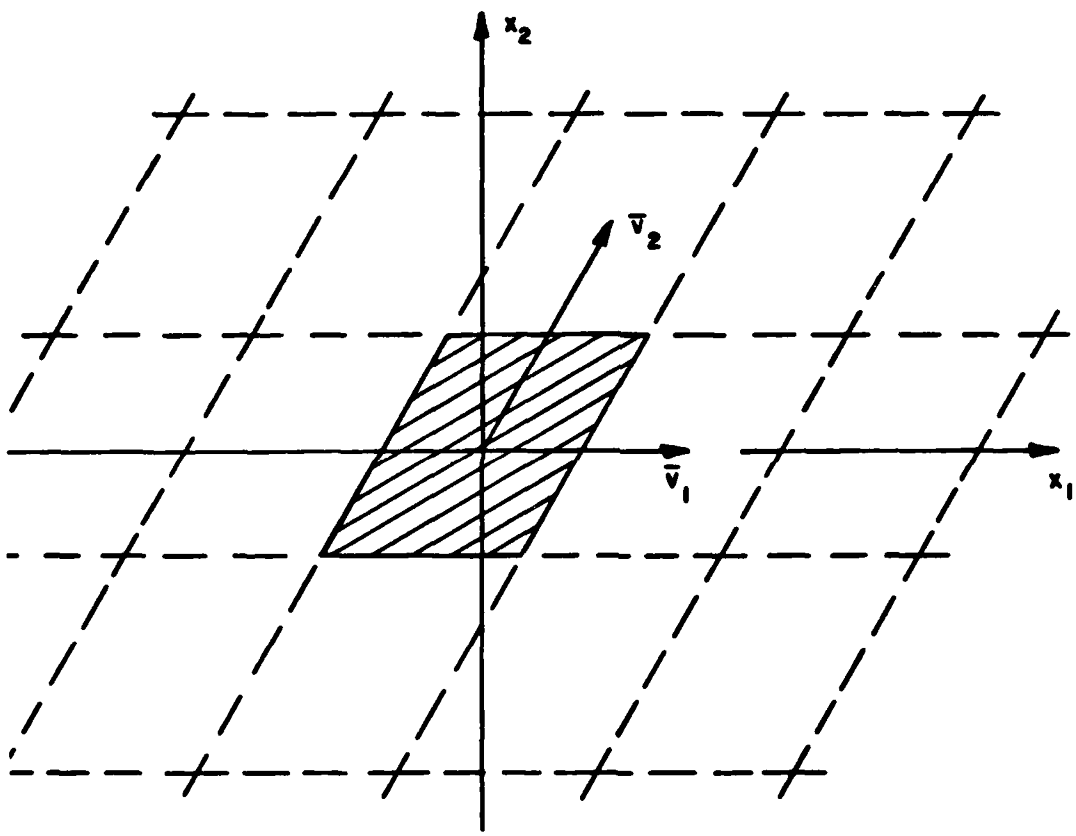


Figure 2-2. Periodicity of the canonical unit: parallelogram (shaded) with its images (dotted lined) defined by $\bar{v}_1$ and $\bar{v}_2$

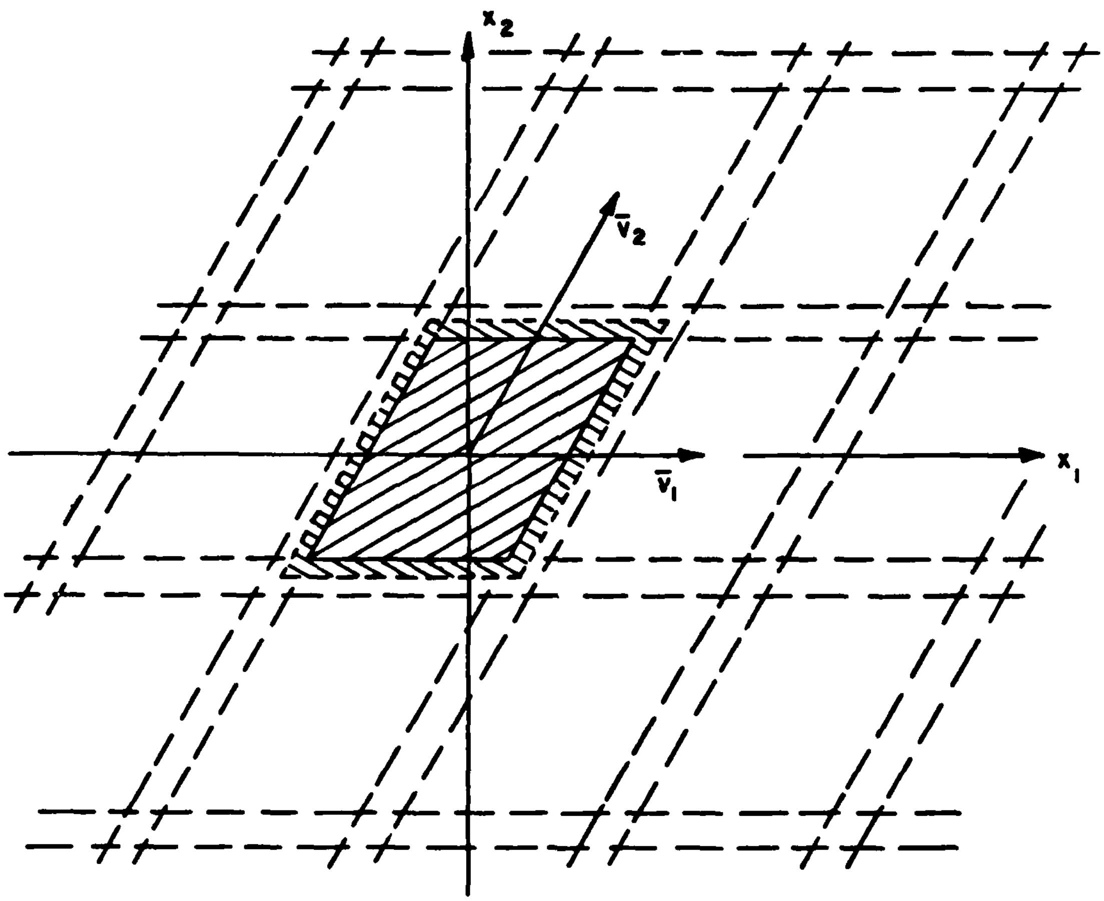


Figure 2-3. New $\bar{v}_1$ and $\bar{v}_2$ defining a similar periodicity as Figure 2-2 except for the extra guard band

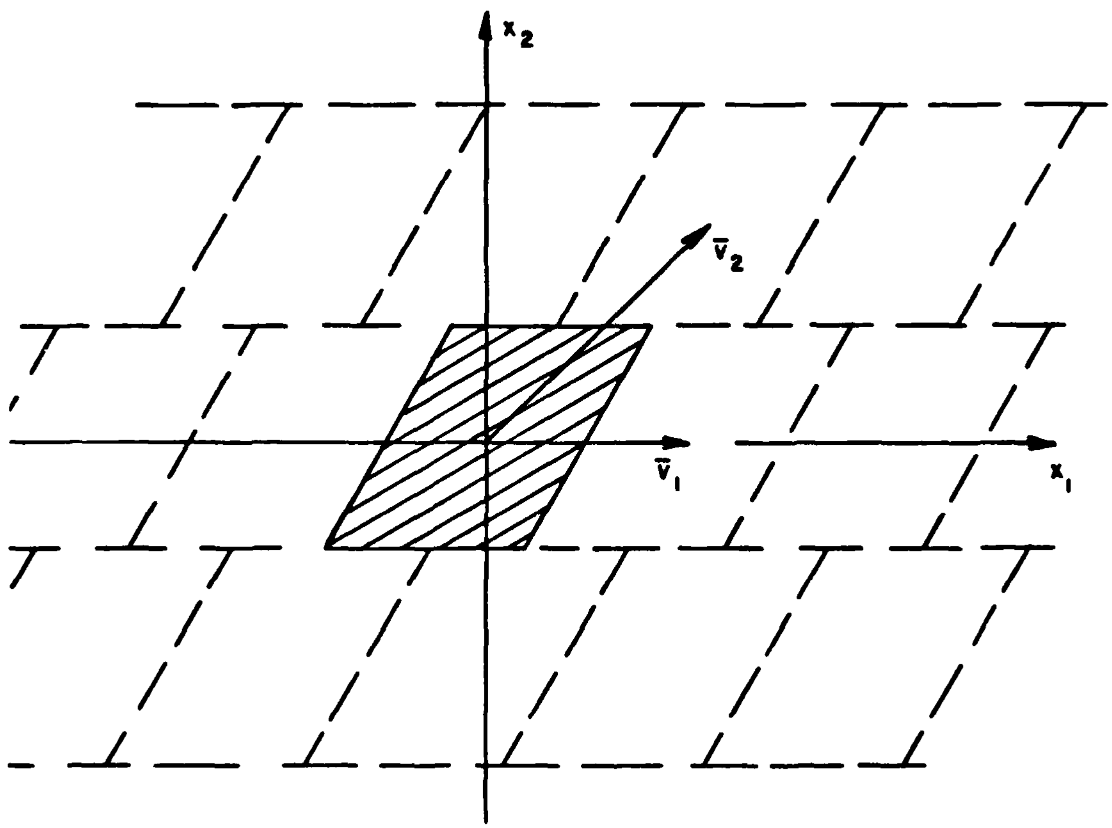


Figure 2-4. New $\bar{v}_1$ and $\bar{v}_2$ defining a similar periodicity as Figure 2-2 except the images are rotated around the unit.

The application of the sampling theorem for the reconstruction of the original signal needs only an interpolation formula, provided the non-overlapping condition is met. In this application, if one has the following:

a) A signal limited in x-space and its images are specified by $\bar{v}_1$, $\bar{v}_2$ with non-overlapping condition met.

b) The sampling lattice in k-space is defined by the vector:

$$l_1\bar{u}_1 + l_2\bar{u}_2, \text{ where } l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots$$

$$\text{with } \bar{v}_i . \bar{u}_j = 2\pi\delta_{ij} : \delta_{ij} = \text{Kronecker delta}$$

or

$$U = 2\pi V^{-T}$$

where

$$U = [\bar{u}_1|\bar{u}_2]$$

$$V = [\bar{v}_1|\bar{v}_2]$$

-T is the notation for the transpose of the matrix inverse of V

c) The interpolation formula

then one can reproduce the two dimensional impulse response. The procedures are:

1) Sample the k-space response at the sampling lattice for $F(l_1\bar{u}_1 + l_2\bar{u}_2)$.

2) Interpolate other required points, if necessary.

The interpolation formula taken from Petersen and Middleton [7].

$$F(\bar{k}_1, \bar{k}_2) = \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} F(l_1\bar{u}_1 + l_2\bar{u}_2)G(k_1\hat{k}_1 + k_2\hat{k}_2 - l_1\bar{u}_1 - l_2\bar{u}_2) \tag{2-9}$$

where $\bar{k}_1 = k_1\hat{k}_1$

$\bar{k}_2 = k_2\hat{k}_2$

3) Inverse Fourier transform the k-space response to obtain the two dimensional impulse response.

$$f(x_1, x_2) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} F(k_1,k_2)e^{j(k_1x_1 + k_2x_2)}dk_1dk_2 \tag{2-10}$$

With the ease of calculation in mind, the periodicity of the object cell and its images are defined as in Figure 2-5 for this report. Its corresponding sampling lattice is shown in Figure 2-6.

The reconstruction function for the parallelogrammatic sampling [7]:

$$G(\omega_1, \omega_2) = \left(\frac{\sin \pi\omega_1}{\pi \omega_1}\right) \left(\frac{\sin \pi\omega_2}{\pi \omega_2}\right) \tag{2-11}$$

where

ω_1 is in the direction of $\bar{u}_1$

ω_2 is in the direction of $\bar{u}_2$

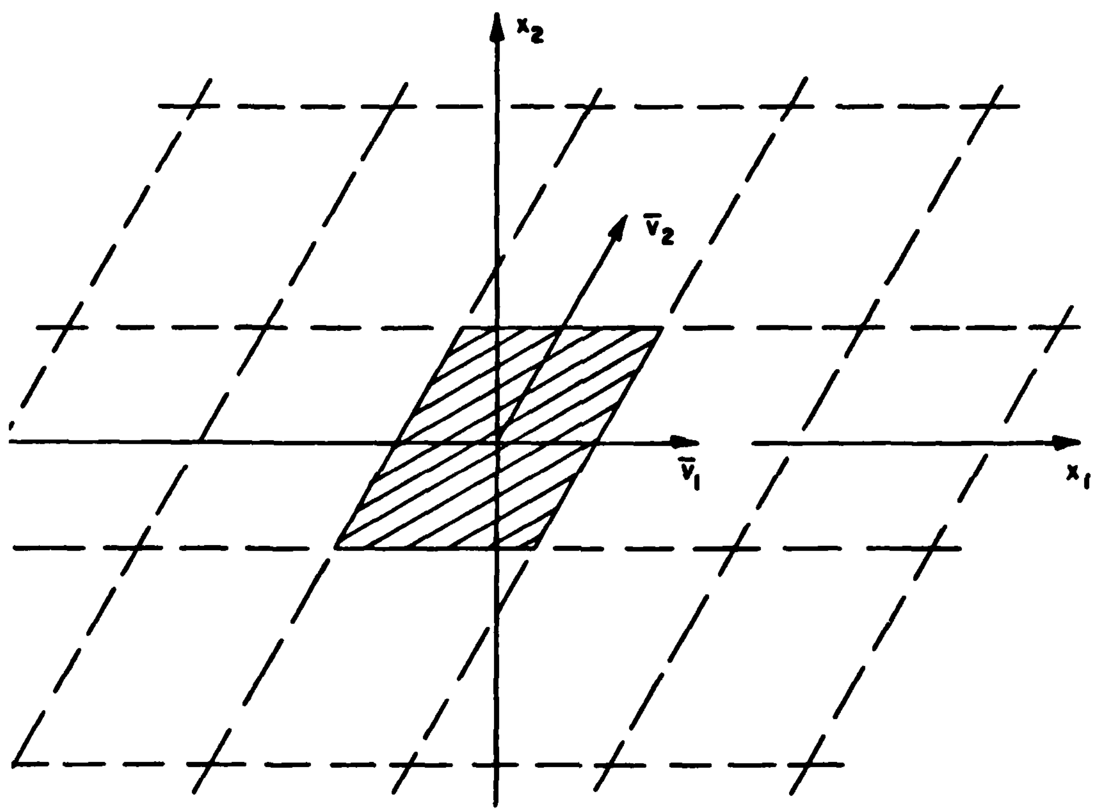


Figure 2-5. A choice of periodicity for the parallelogrammatic containment unit

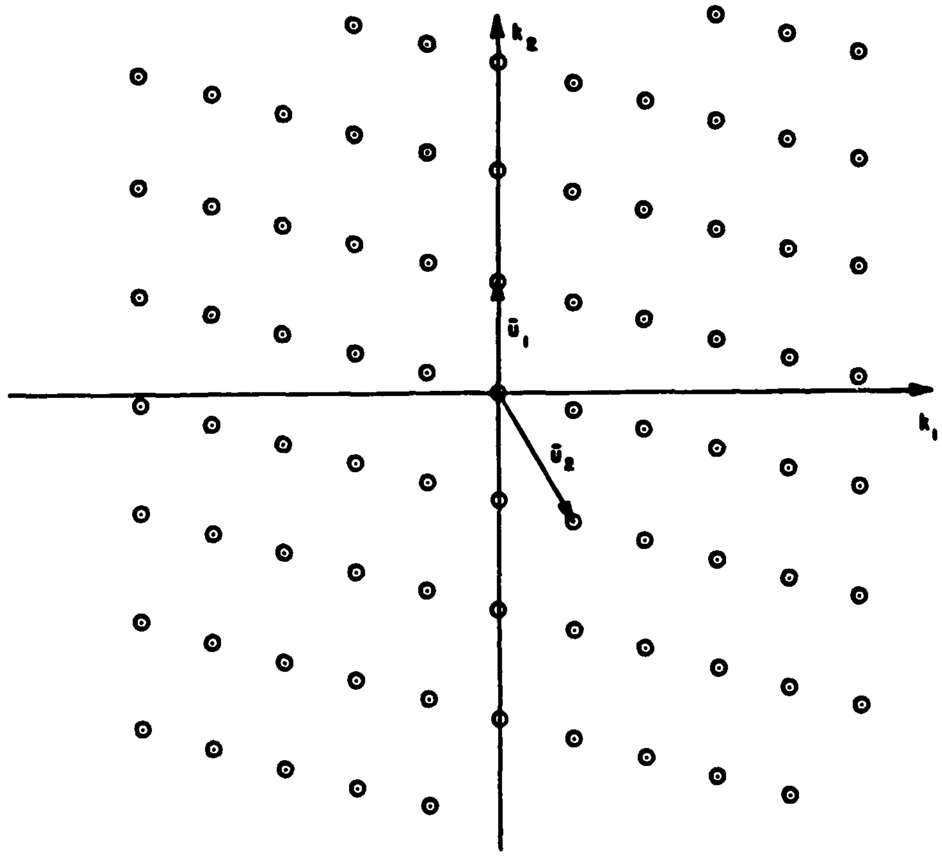


Figure 2-6. Sampling lattice defined by the choice of periodicity in Figure 2-5

For the sphere or isotropic confinement, the configuration is adopted from the concept of closest packing of spheres by Coxeter [9]. The configuration chosen for this report is as shown in Figure 2-7. The corresponding sampling lattice is shown in Figure 2-8.

For the two dimensional case [7,9]:

$$\bar{v}_1 = R\begin{bmatrix} \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{bmatrix} \qquad \bar{v}_2 = R\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\bar{u}_1 = (2\pi/R)\begin{bmatrix} \frac{2}{\sqrt{3}} \\ 0 \end{bmatrix} \qquad \bar{u}_2 = (2\pi/R)\begin{bmatrix} \frac{1}{\sqrt{3}} \\ 1 \end{bmatrix} \tag{2-12}$$

where R is the radius of the isotropic cell. The reconstruction function [7]:

$$G(\omega_1, \omega_2) = \frac{1}{R^2\omega_1(\omega_1^2 - 3\omega_2^2)} \times \{ 2\omega_1\cos(R\omega_1/\sqrt{3})\cos(R\omega_2) - 2\omega_1\cos(2R\omega_1/\sqrt{3}) - 2\sqrt{3}\omega_2\sin(R\omega_1/\sqrt{3})\sin(R\omega_2) \} \tag{2-13}*$$

where

ω_1 is in the direction of $\bar{u}_1$

ω_2 is in the direction of $\bar{u}_2$

(* There is a ω_2 factor missing in the third term of this expression in reference [7]. See Appendix A for details of the derivation).

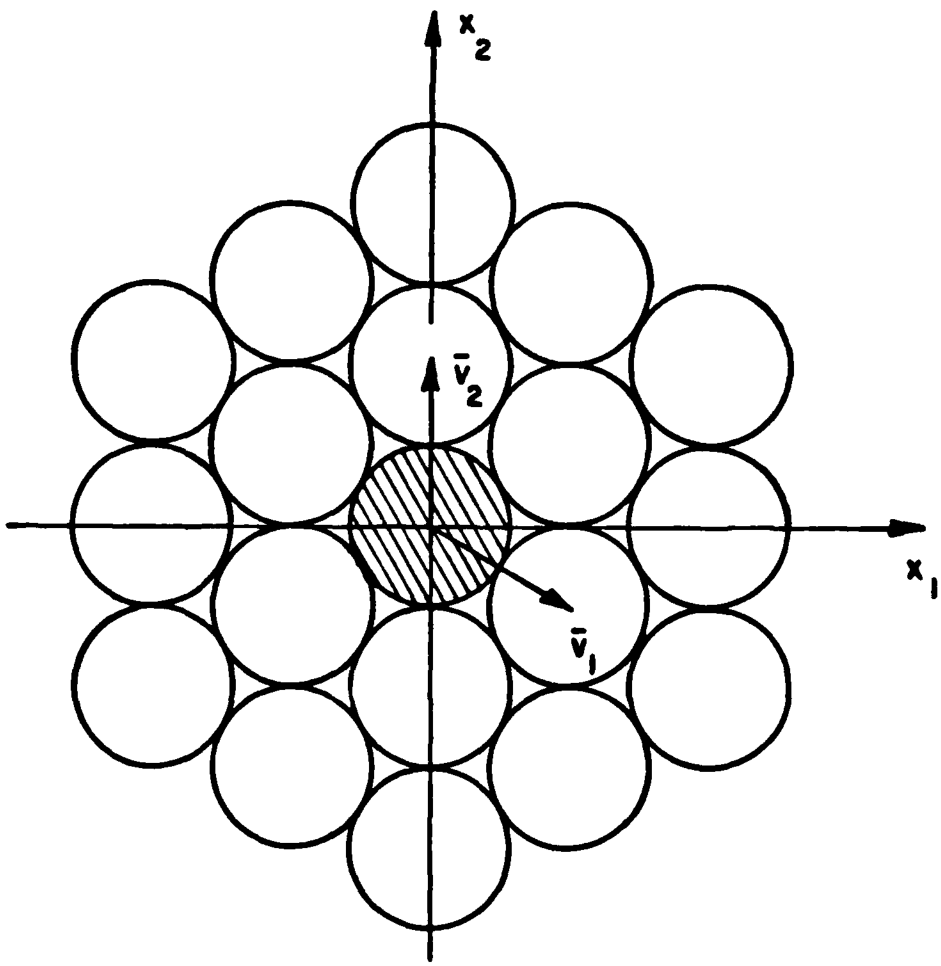


Figure 2-7. A choice of periodicity for the circular containment cell

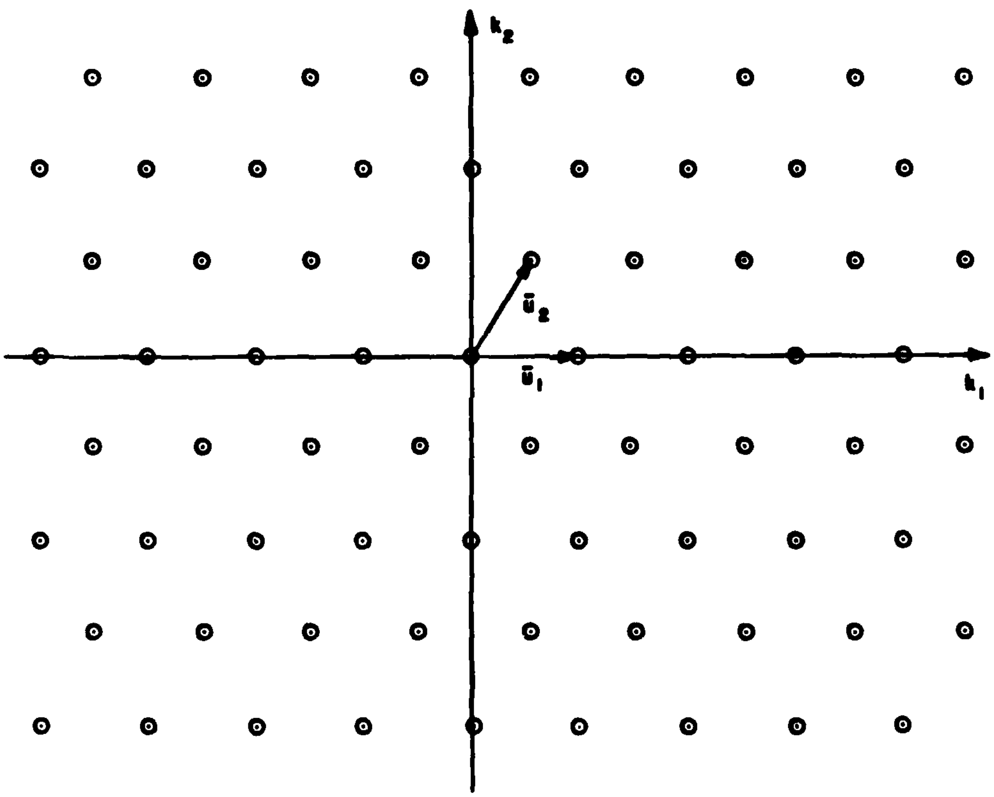


Figure 2-8. Sampling lattice defined by the choice of periodicity in Figure 2-7

Since this report only deals up to two dimensional sampling, the reader is referred to Petersen and Middleton's paper on the three dimensional reconstruction function $G(\omega_1, \omega_2, \omega_3)$.

The N dimensional sampling theorem gives the minimum sampling lattice or criterion which will sufficiently define the k-space signal. All other non sampled k-space values can be interpolated via an extension of Equation 2-9. The k-space signal can be inverse Fourier transformed into the spatial domain to give a representation of the spatial impulse response. In another words, the spatial signal is also characterized by those k-space lattice samples.

If one is interested only in the two dimensional impulse response, then Mensa et al. [8] presents a different view on the sampling criterion. Unfortunately, it is only applicable to the two dimensions. First, rectangular to polar coordinate transformation is applied to the two dimensional Fourier integral (Equation 2-10):

$$f(x_1, x_2) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} F(k_1, k_2) e^{j(k_1 x_1 + k_2 x_2)} dk_1\, dk_2 \qquad (2\text{-}14)$$

$$\left\{ \text{Let} \quad \begin{aligned} r^2 &= k_1^2 + k_2^2 \quad ; \quad \phi = \tan^{-1}(k_2/k_1) \\ \rho^2 &= x_1^2 + x_2^2 \quad ; \quad \theta = \tan^{-1}(x_2/x_1) \end{aligned} \right\} \qquad (2\text{-}15)$$

$$= \frac{1}{4\pi 2} \int_{r=o}^{\infty} \int_{\phi=o}^{2\pi} r\tilde{F}(r, \phi)e^{j2\pi r\rho\cos(\phi-\theta)}dr\, d\phi \tag{2-16}$$

Then the double integral is reduced to a single integral by considering only a particular frequency ring (i.e. $\delta(r-\omega_i)$).

$$\tilde{f}_i(\rho, \theta) = \frac{\omega_i}{4\pi 2} \int_{\phi=o}^{2\pi} \tilde{F}(\omega_i, \phi)e^{j2\pi\omega_i\rho\cos(\phi - \theta)} d\phi \tag{2-17}$$

Thus, the two dimensional Fourier transformation is reduced to a convolution type integral. $F(k_1, k_2)$ is the frequency response of the target in the two dimensional k-space. The notion of $\delta(r-\omega_i)$ represents information taken only with one frequency. Subsequently, the two dimensional impulse response is obtainable via one integration. If information from other frequency rings are available, then superposition of every frequency ring response in the spatial domain will give a wide band two dimensional impulse response representation.

$$f_T(x_1, x_2) = \sum_i \tilde{f}_i(\rho, \theta) \tag{2-18}$$

where

$x_1 = \rho\cos\theta$

$x_2 = \rho\sin\theta$

Naturally, the Nyquist spacings between the frequency rings and between angular samples must be used, before the total two dimensional time response obtained can be considered a sufficient representation of the true two dimensional time response.

The angular increment which satisfies the Nyquist criterion is given by Mensa et al. [8]:

$$\Delta\theta \begin{cases} < \dfrac{\lambda}{2D} & \lambda \ll 2D \\ < \sin^{-1}\left(\dfrac{\lambda}{2D}\right) & \lambda < 2D \\ < \dfrac{\pi}{4} & \lambda > 2D \end{cases} \tag{2-19}$$

where D = maximum dimension of the object

λ = wavelength of the frequency to be used

(Assumption: The origin of the x-space coincides with the middle of the object.)

The author would like to propose the following for the frequency increment:

$$\Delta f < \frac{c}{2(1 + K)D} \tag{2-20}$$

where D = maximum dimension of the object

c = speed of light

K = some safety factor

The aspect angle which has the longest dimension of the object is assumed to have the longest settling time in its impulse response. All other aspect angles require Nyquist frequency increments larger than or equal to this aspect angle. The value of the safety factor is the best estimation achievable by other means. The reason for the safety factor is not all impulse response signals are limited to the length of the object at any aspect angle. Furthermore, in the GTD sense, some of the

multiple scattering effects may be eliminated or included by employing a smaller or larger value of the safety factor. This is equivalent to truncation of the signal in time during measurement.

CHAPTER III

PRACTICAL CONSIDERATIONS

In this chapter, the practical aspects of the previously described theory are presented. Since negative frequencies cannot be physically generated, an assumption on the negative frequency response must be made. First, the assumption of a real measured signal is discussed. The frequency limits are then considered in relation to the object's size, its impulse response and the data processing requirement. For clarity, the practical aspects are discussed in one dimension. Extension to the higher dimensions can be easily accomplished.

In general, a true impluse is hard to generate; instead, a Gaussian pulse is often used. There are also times when good narrow Gaussian pulses are not readily available. In these cases, the approach of frequency sweeping may be used. The bandwidths of most oscillators, waveguide components, transmitting and receiving antennas are limited. In essence, the frequency band can only be swept from ω_L to ω_H (Figure 3-1). To overcome part of the problem, let's consider the one dimensional sampling theorem [1, 2]:

Figure 3-1. Frequency information available

If $f(t) = 0 \qquad \text{for } |t| \geq T$

then $F(\omega)$ can be uniquely determined from

$$F_n = F(\frac{n\pi}{T})$$

and

$$F(\omega) = \sum_{n=-\infty}^{\infty} F(\frac{n\pi}{T}) \frac{\sin(\omega T - n\pi)}{\omega T - n\pi} \tag{3-1}$$

{Note: If $F(\omega)$ is even, then the required measurement information is $F(\frac{n\pi}{T})$; for $n = 0, 1, 2, \ldots$}

Consider the inverse Fourier Transform

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega \tag{3-2}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\cos(\omega t)\, d\omega + \frac{j}{2\pi} \int_{-\infty}^{\infty} F(\omega)\sin(\omega t)\, d\omega \tag{3-3}$$

If $F(\omega)$ is even, then the integrand in the second integral is an odd function of ω. It follows that the second integral is zero. Therefore,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\cos(\omega t)\, d\omega \tag{3-4}$$

$$= \text{real function, if } F(\omega) \text{ is even}$$

(Note: If $F(\omega)$ is complex, then $Re[F(\omega)] = Re[F(-\omega)]$ and $Im[F(\omega)] = -Im[F(-\omega)]$ are the conditions for $f(t)$ to be real).

In this discussion, the object is real. It follows that $f(t)$ is also real. Now, information from $-\omega_H < \omega < -\omega_L$ and $\omega_L < \omega < \omega_H$ is available. (Figure 3-2)

Next, the Rayleigh Law is used to determine the scattered field at d.c. or zero frequency. From Kennaugh and Cosgriff [3]: "As the source frequency tends to zero, all finite scatterers follow the Rayleigh Law, giving a scattered field intensity which diminishes as the square of frequency." The scattered field intensity at d.c. will be zero. If

$$\frac{\pi}{T} > \omega_L,$$

then

$$F_n \text{ is known for } -N < n < N$$

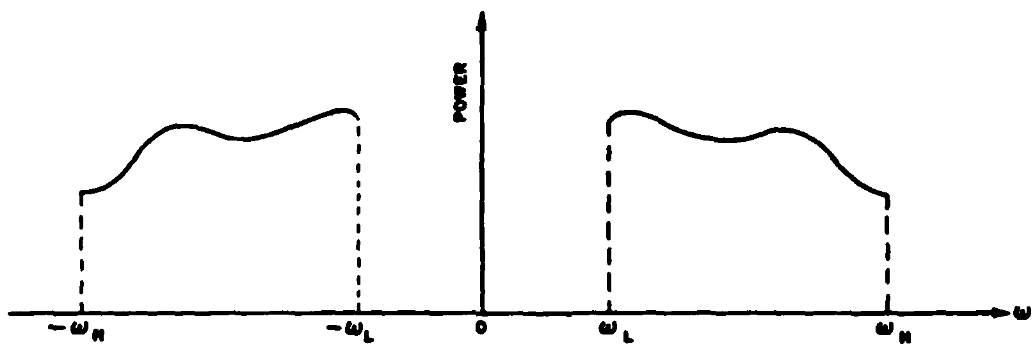


Figure 3-2. Frequency information available after assumption is made about the negative frequency samples

where

$$N = I\left[\frac{\omega_H T}{\pi}\right] \tag{3-5}$$

$I[x]$=truncation of x's value after the decimal point

That is, if the first sampled location $(\frac{\pi}{T})$ is higher than or equal to ω_L, then information is available from $-\omega_H$ to ω_H, because one can interpolate the in between data points using the sampling theorem.

Let

x_s be the object size

K be some safety factor

then the settling time T_s is

$$T_s = 2(1+K)\frac{x_s}{c} \quad \text{with } T = T_s/2$$

One must have the condition:

$$\frac{\pi}{T} > \omega_L$$

$$\frac{2\pi c}{2(1+K)x_s} > \omega_L$$

$$\Leftrightarrow \quad \frac{\pi c}{(1+K)x_s} > \frac{2\pi c}{\lambda_L}$$

$$\Leftrightarrow \quad \lambda_L > 2(1+K)x_s \tag{3-6}$$

This puts a limit on the lowest frequency usable, or the largest object size assumed by a given ω_L.

The span $(\omega_H - \omega_L)$ effectively determines the resolution of the *impulse response signature. Let x_r be the desired resolution on the* object, then the impulse response resolution is

$$t_r = \frac{2x_r}{c}$$

$$\Longleftrightarrow \qquad \omega_r = \frac{2\pi c}{2x_r}$$

One has the condition:

$$\omega_H > \frac{2\pi c}{2x_r} = \omega_r$$

$$\Longleftrightarrow \frac{2\pi c}{\lambda_H} > \frac{\pi c}{x_r}$$

$$\Longleftrightarrow \lambda_H < 2x_r \tag{3-7}$$

Conditions 3-6 and 3-7 will help to decide how wide a frequency band may be used. If the bandwidth is wide enough, then the impulse response can be generated to a very good approximation.

A common problem during implementation may involve the inverse Fourier transform on the reconstructed frequency spectrum. This waveform may or may not be a well defined function which can be inverse Fourier transformed into a closed form solution. The common approach would be to approximate the integration using a summation on a digital computer. In effect, this approach will be a Fourier series representation which requires the time or frequency waveform to be periodic. Consequently, there are 2 more limitations:

1) The period (T_c) used in the digital computation must be greater than two times the settling time (T_s) of the impulse response. (i.e., $T_c > 2T_s$)

2) The period (ω_c) used in the digital computation must be greater than two times the highest frequency (ω_H) swept. (i.e., $\omega_c > 2\omega_H$)

Fortunately, a standard IBM subroutine FFT (Fast Fourier Transform) package is available to do the required Fourier analyses.

Furthermore, one dimensional impulse response requires a two dimensional plot. The two axis quantities are amplitude and time. Two dimensional impulse response requires a three dimensional plot. The three axis quantities are the amplitude and the plane axes. Should anyone consider three dimensional impulse response, one requires a plot in four dimensional space. Consequently, this report will only deal with impulse responses up to two dimensions. With the knowledge in one's mind that the spatial impulse response can easily be obtained by an extension of this two dimensional approach when there is an appropriate representation.

CHAPTER IV

ONE DIMENSIONAL IMPULSE RESPONSES

In this chapter, the interpolation of one dimensional impulse responses is presented; first, the result of interpolation using one dimensional data; then, using two dimensional data. The object is a six inch diameter metallic sphere. This object choice is because the Mie solution in frequency domain is readily available. In this first section, the Mie solution frequency data are sampled at different rates and interpolated either using a straight line or a sinc reconstruction function.

$$F(\omega) = \sum_{n=-N}^{N} F\left(\frac{n\pi}{T}\right) \frac{\sin(\omega T - n\pi)}{\omega T - n\pi} \tag{4-1}$$

where

$$N = I\left(\frac{\omega_H T}{\pi}\right)$$

$I(x)$ = truncation of x's value after the decimal point

ω_H = highest frequency used

[Note: This is Equation (3-1) except the summation is from -N to + N].

A cosine tapering weighting function [10]: (Figure 4-1) (for the convenience of the reader, all tables and figures of Chapter IV are grouped together at the end of the chapter)

$$W(n)\begin{cases} = 0.5[1+\cos(\frac{\pi n}{N_1})] & |n| < N_1 \\ = 0 & |n| > N_1 \end{cases} \tag{4-2}$$

is multiplied to the interpolated frequency data. The purpose of the weighting or filtering is to reduce the effect of the Gibb's phenomenon [2]. The resulting data are inverse Fourier transformed into the time domain discretely to give the impulse response in time.

Figure 4-2 represents a time domain impulse response plot obtained from the Mie solution for a six inch diameter metallic sphere using the frequency spectrum from 0 to 12 Ghz with a cosine tapering filter. The imperfect specular impulse and the ripples around it at the start of the response are caused by 1) finite bandwidth, and 2) Gibbs' phenomenon.

Figure 4-2 is considered to be the standard for comparison with the other responses. Its frequency samples are taken every 60 Mhz and linked together by straight lines. Figure 4-3 has frequency samples taken every 0.125 Ghz over the spectrum of 0 to 12 Ghz and interpolated the in between points using Equation (4-1). The differences of the impulse responses in this chapter from Figure 4-2 (the 'exact' solution) are shown in figures designated with their respective figure number plus an 'a' attached. For example, to obtain Figure 4-2 from Figure 4-3, one adds Figure 4-3a to Figure 4-3.

Figures 4-4, 4-5, and 4-6 are similar to Figure 4-3, except frequency samples are taken every 0.25 Ghz, 0.5 Ghz, and 0.75 Ghz respectively. Figure 4-5 is still recognizable to be Figure 4-2 but Figure 4-6 is not. Figure 4-6 does not have similar behavior because it employs a frequency sampling rate below the Nyquist rate. From Figure 4-2, one can estimate the settling time of the impulse response of the sphere to be about 1.5 nsec. Equally well, one could use a longer settling time depending on one's assumption of the noise amplitude. i.e.,

$$T = 0.75E{-}9s$$

$$f_s > \frac{1}{2T} = \frac{1}{1.5E{-}9s}$$

$$\cong 0.66 \text{ Ghz}$$

$$< 0.75 \text{ Ghz (Figure 4-6)}$$

Figures 4-2 to 4-5 have sampling rate better than the Nyquist rate. If the sampling rate is high enough, then one may use straight line interpolation (Figure 4-2) instead of the sinc reconstruction function to save computer time on interpolation. On the other hand, if samples are scarce but still satisfy the Nyquist criterion, then the sinc reconstruction function (Equation (4-1)) is preferred for more pleasing results. (Figures 4-3, 4-4, 4-5)

Now, the reconstruction of the one dimensional impulse responses from data obtained on two dimensional isotropic and cubic sampling lattice is presented.

$$F(\bar{k}_1, \bar{k}_2) = \sum_{l_1} \sum_{l_2} F(l_1\bar{u}_1 + l_2\bar{u}_2)G(k_1\hat{k}_1 + k_2\hat{k}_2 - l_1\bar{u}_1 - l_2\bar{u}_2) \quad (4\text{-}3)$$

where

l_1 and l_2 are summed over l_1 and l_2 which satisfy

$k_L < |l_1\bar{u}_1 + l_2\bar{u}_2| < k_H$

and

k_L and k_H define the frequency bandwidth used.

(Note: This is Equation (2-9) with finite summation.)

Samples are taken out of the Mie's frequency solution on the isotropic sampling lattice defined by $\bar{u}_1$ and $\bar{u}_2$ in Equation (2-12), and the cubic sampling lattice defined by :

$$\bar{u}_1 = 2\pi \begin{bmatrix}1\\0\end{bmatrix} \qquad \bar{u}_2 = 2\pi \begin{bmatrix}0\\1\end{bmatrix} \tag{4-4}$$

i.e.,

$$\text{samples are taken over } l_1\bar{u}_1 + l_2\bar{u}_2 \text{ for } l_1, l_2 = 0, \pm 1, \pm 2, \ldots \tag{4-5}$$

This sampling lattice is generated by the program PTGRID (see Appendix B). The frequency response for an aspect angle is interpolated using Equation (4-3) in conjunction with either Equation (2-13) for isotropic sampling or Equation (2-11) for cubic sampling. This interpolation work is done by the program INTERPOL (see Appendix B). The resulting frequency response is again cosine tapering low pass filtered to reduce the Gibb's phenomenon. After the filtering, the frequency spectrum is inverse Fourier transformed discretely into the time domain to give an impulse response picture for the six inch metallic sphere. The

filtering and the discrete inverse Fourier transform are functions of FTRAN [11] (see Appendix B).

Sampling at every 0.5 Ghz for a six inch diameter metallic sphere, is equivalent to assuming a signal having four times the diameter of the sphere in one dimensional space. Therefore, the two dimensional confinement cell is assumed to contain the sphere and has a guard band of 1.5 times the maximum dimension of the sphere surrounding the sphere. The safety factor thus chosen is 1 (i.e., $2(1+K)=4 \Leftrightarrow K=1$). The frequency range sampled is 0 to 12 Ghz.

Figures 4-7, 4-8, 4-9, and 4-10 are one dimensional impulse responses reconstructed from two dimensional isotropic sampling data, for aspect angles of 0, 0.719, 1.438, 30 degree respectively. The CPU time taken for interpolating each waveform is about 5 minutes for 200 plotting points. Figures 4-11, 4-12, 4-13, 4-14 are reconstructed from cubic sampling data, for aspect angles of 0, 1.193, 2.386, 45 degrees. The CPU time taken for these waveforms is about 2.5 minutes for 200 plotting points. Less time in interpolation for cubic sampled data is probably due to the simplicity of the reconstruction function (Equation (2-11) versus (2-13)). These angular choices are arbitrarily chosen. One should note the close resemblance of all these figures (4-7 to 4-14) with Figure 4-5. Next, let's consider the case of more samples taken. Effectively, the safety factor is changed from one to three but the frequency range remains the same. Figures 4-15, 4-16, and 4-17 are reconstructed one dimensional impulse responses using isotropic samples for aspect angles of 0, 0.352, 30 degrees respectively. Again discrepancy is not high (Figures 4-15a, 4-16a, and 4-17a). Since the

density of samples taken is finer, or more samples participated in the interpolation, the CPU time has incresed to about 20 minutes for 200 plotting points.

If both the isotropic and cubic sampling can perform competitively, how does one decide on which sampling grid? The answer lies in the efficiency definition defined previously. However, the area is considered in a plane instead of the volume in a three dimensional space.

i.e., one modifies Equations (2-6) and (2-7) to

$$\text{Efficiency} = \frac{\text{AREA OF A CROSS-SECTION ON THE OBJECT}}{\text{AREA OF THE SAME CROSS-SECTION ON THE TWO DIMENSIONAL ENCLOSURE}} \tag{4-6}$$

but the decision rule: Equation (2-8), remains the same.

Efficiency, as mentioned before, is defined as a minimum sampling requirement. Three objects: a six inch diameter sphere, a $3\sqrt{2}$ inch cube and a sphere cap cylinder are shown in Figure 4-18 on their major axis cross-section. Their respective cross-sectional areas; closest circular, squared, rectangular enclosure cross-sectional areas; and efficiencies are tabulated in Table 4-1.

The definition of Equation (4-5) is used to locate the sampling lattice. The number of these locations over a frequency range is summed to give the minimum number of sampling, defined by the sampling theorem, to sufficiently characterize the spatial impluse response in that frequency range. This work is done by the program PTGRID (see Appendix B). The numbers are tabulated in Table 4-2 and plotted in Figures 4-19,

4-20, and 4-21. A safety factor of one is used in the computation. The sizes of the objects are chosen such that the circular enclosure is the same for all three objects for easy comparison in the graphs.

As the frequency range becomes larger and larger, the significance of the efficient sampling grid becomes more and more important. Let's take the example of the sphere cap cylinder (Figure 4-19). The use of the rectangular enclosure will provide an efficiency of 0.96. The number of samples required over 0 to 12 Ghz is 696. The use of the squared enclosure can only give an efficiency of 0.38. The number of samples required is 2.5 times that of the rectangular enclosure. The use of the circular enclosure has an efficiency of 0.45. The number of samples is about 2.3 times that of the rectangular enclosure. As the frequency range is expanded, saving in measurement time by the proper choice of the enclosure becomes very substantial.

TABLE 4-1

EFFICIENCY COMPARISON ON 3 OBJECTS USING 3 TYPES OF CONTAINMENT CELLS

Object	SPHERE	CUBE	SPHERE CAP CYLINDER
Area on major axis	182.4	116.1	82.2
Area of closest circular enclosure	182.4	182.4	182.4
Area of closest squared enclosure	232.3	116.1	214.7
Area of closest rectangular enclosure	232.3	116.1	85.9
Efficiency of circular enclosure	1	0.64	0.45
Efficiency of squared enclosure	0.79	1	0.38
Efficiency of rectangular enclosure	0.79	1	0.96

(Note: the areas are in units of squared centimeters)

TABLE 4-2

THE NUMBER OF SAMPLING POINTS FOR 3 TYPES OF SAMPLING
ON 3 OBJECTS OVER VARIOUS FREQUENCY RANGES

Object	SPHERE		CUBE	SPHERE CAP CYLINDER	
Type of enclosure	circular	squared	squared	rectangular	squared
Frequency ranges from 0 up to (Ghz)					
1	12	12	8	2	8
2	42	48	24	20	44
3	96	120	60	46	108
4	186	212	100	80	192
5	282	324	160	116	292
6	408	472	240	174	436
7	558	640	324	236	592
8	720	828	420	310	768
9	912	1048	516	384	972
10	1134	1304	656	476	1200
11	1368	1564	776	582	1456
12	1626	1876	940	696	1740
Plotted in Figure	4-19	-	-	4-19	4-19
	4-20	4-20	-	-	-
	4-21	-	4-21	-	-
Notation:	x	o	o	□	o

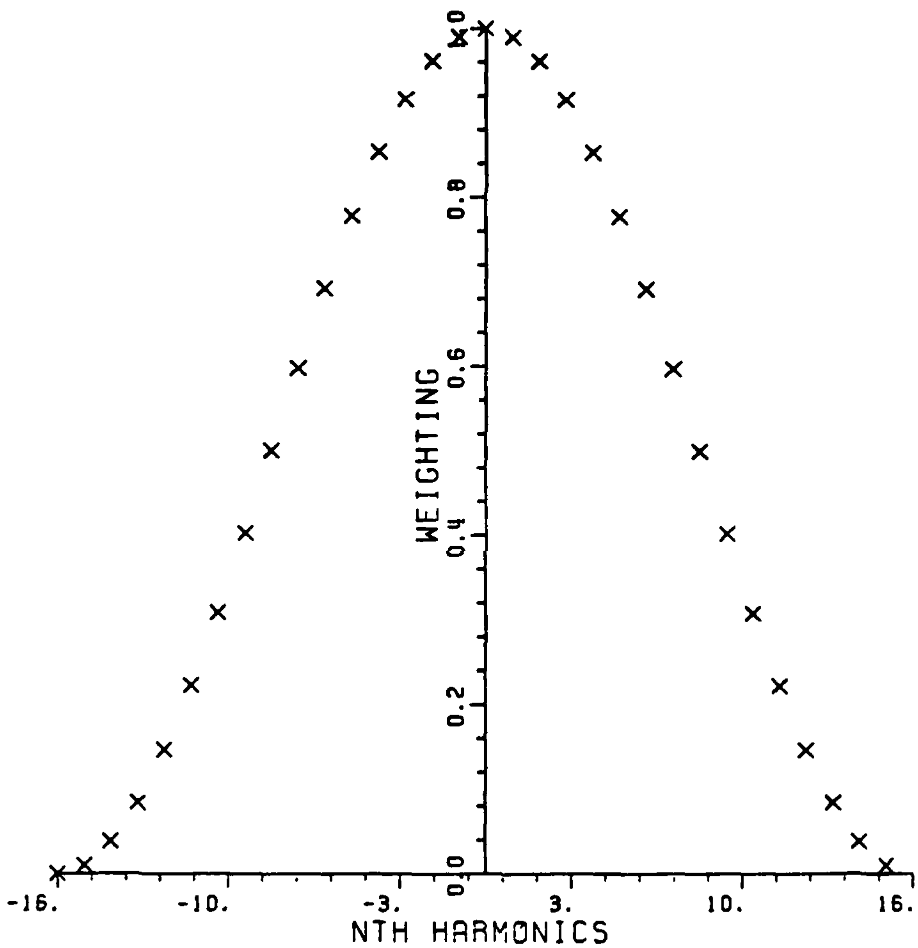


Figure 4-1. Cosine tapering weighting (Equation (4-2) with $N_1 = 16$)

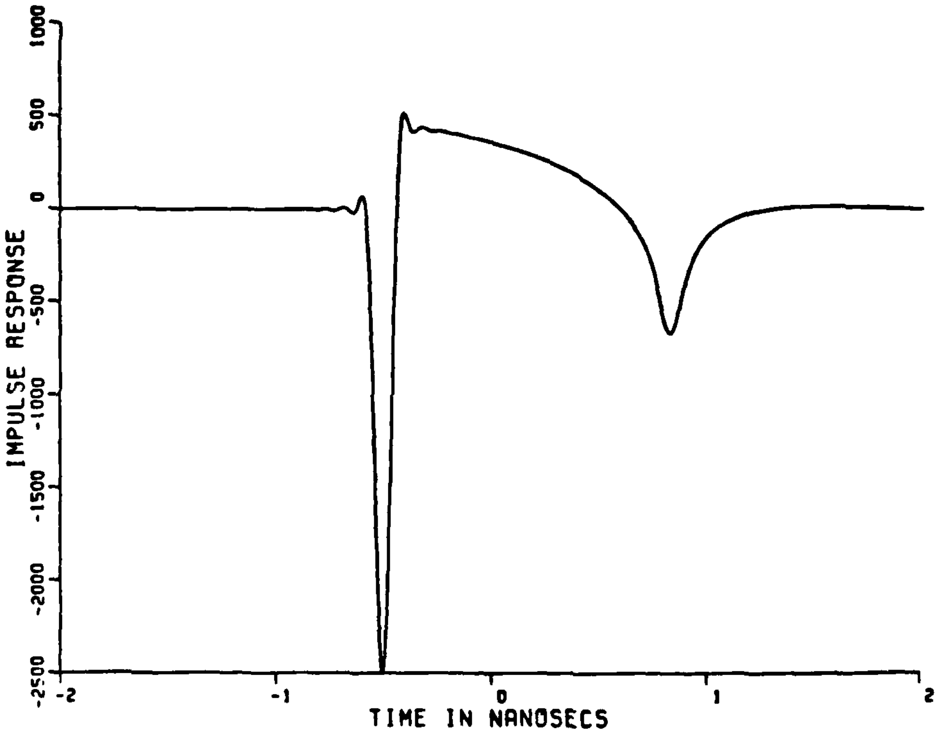


Figure 4-2. Impulse response of a 6" metallic sphere with frequency samples taken every 60 Mhz over the range of 0 to 12 Ghz

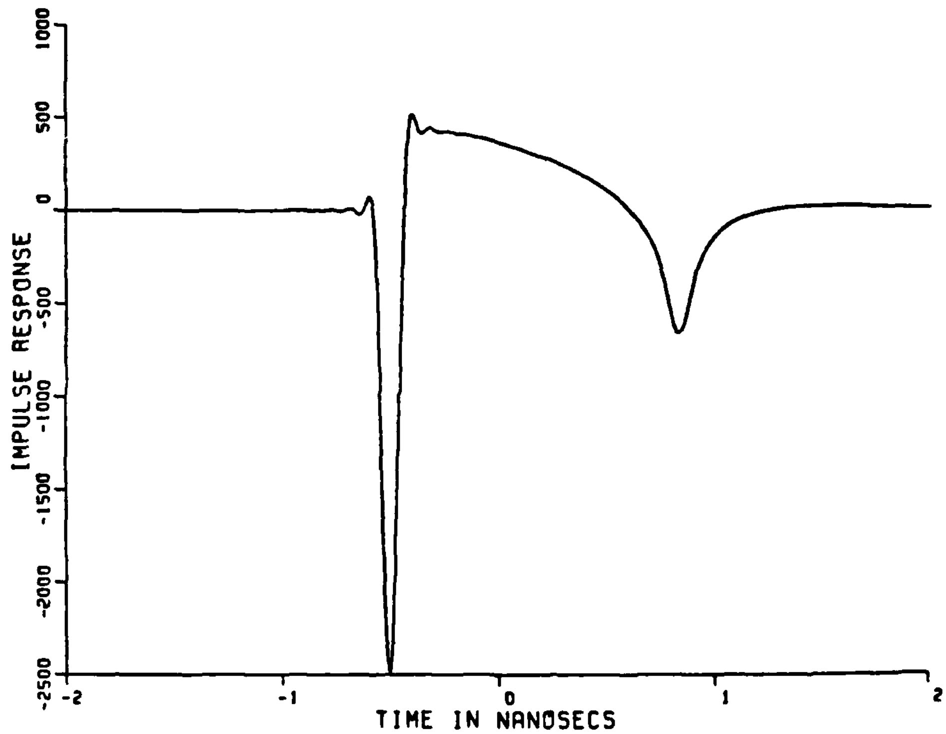


Figure 4-3. Impulse response of a 6" metallic sphere with frequency samples taken every 125 Mhz over the range of 0 to 12 Ghz and interpolated using Equation (4-1)

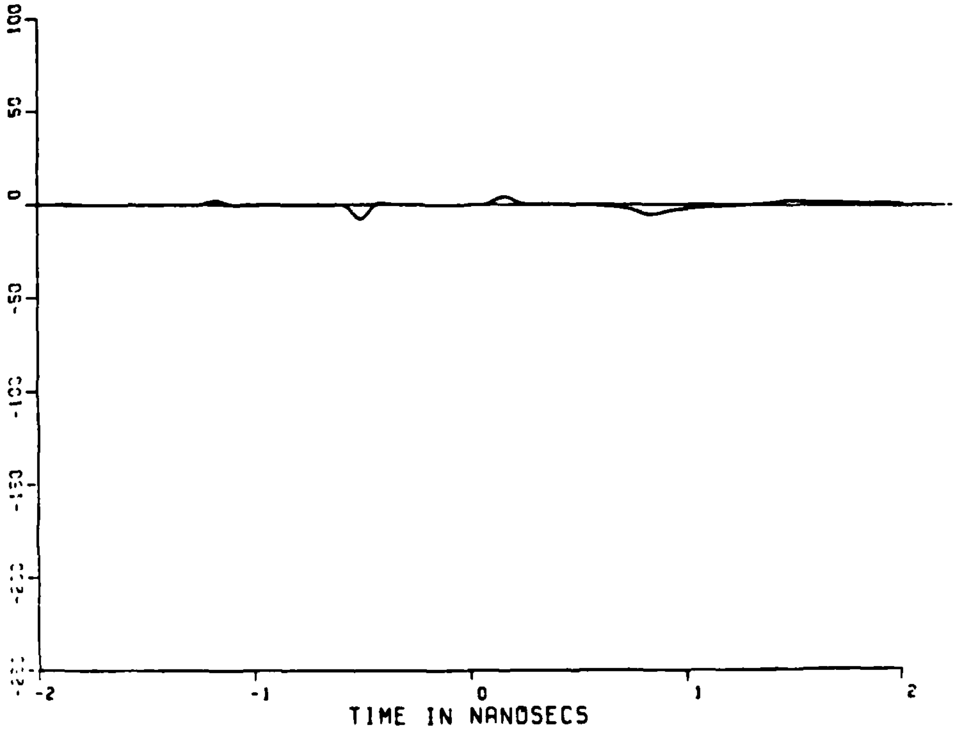


Figure 4-3a. Error of Figure 4-3 from Figure 4-2
(Magnification = 10x)

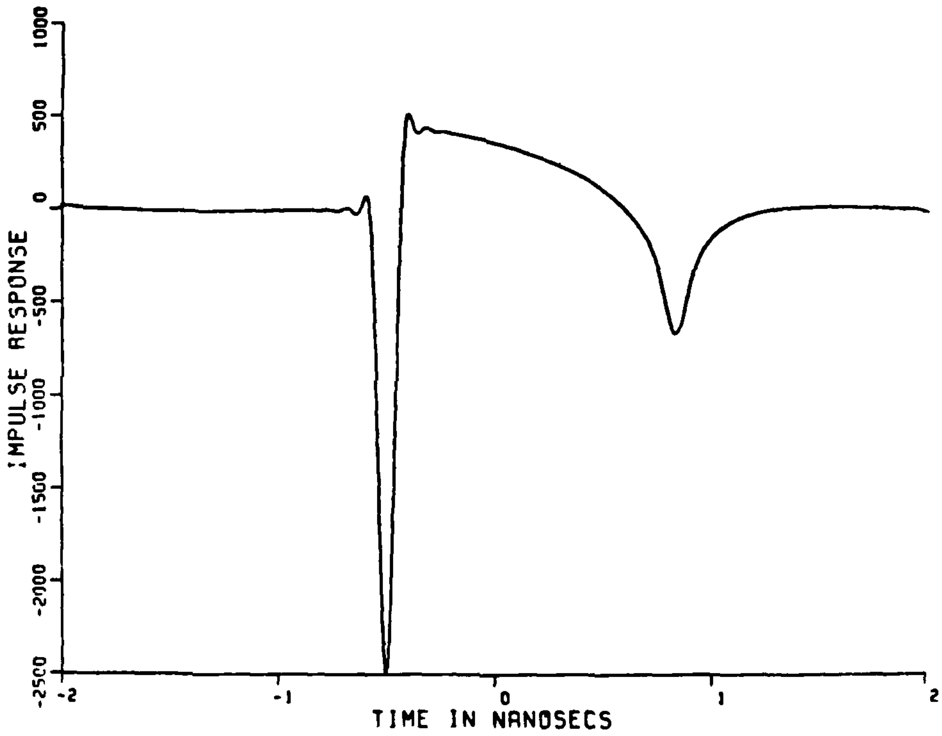


Figure 4-4. Similar description as Figure 4-3, except frequency samples are taken every 250 Mhz

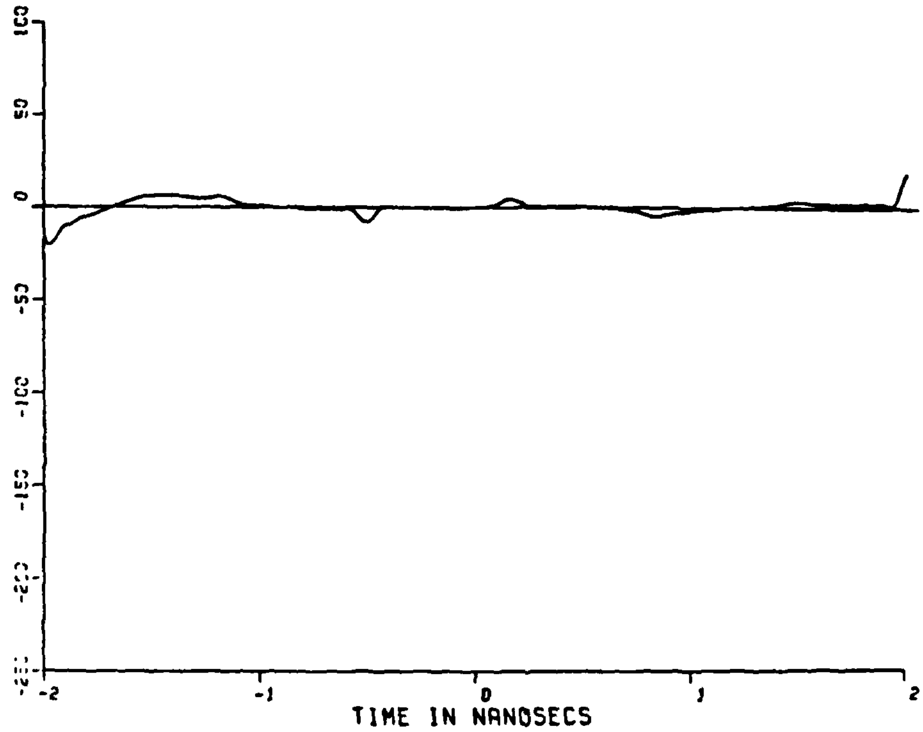


Figure 4-4a. Error of Figure 4-4 from Figure 4-2
(Magnification = 10x)

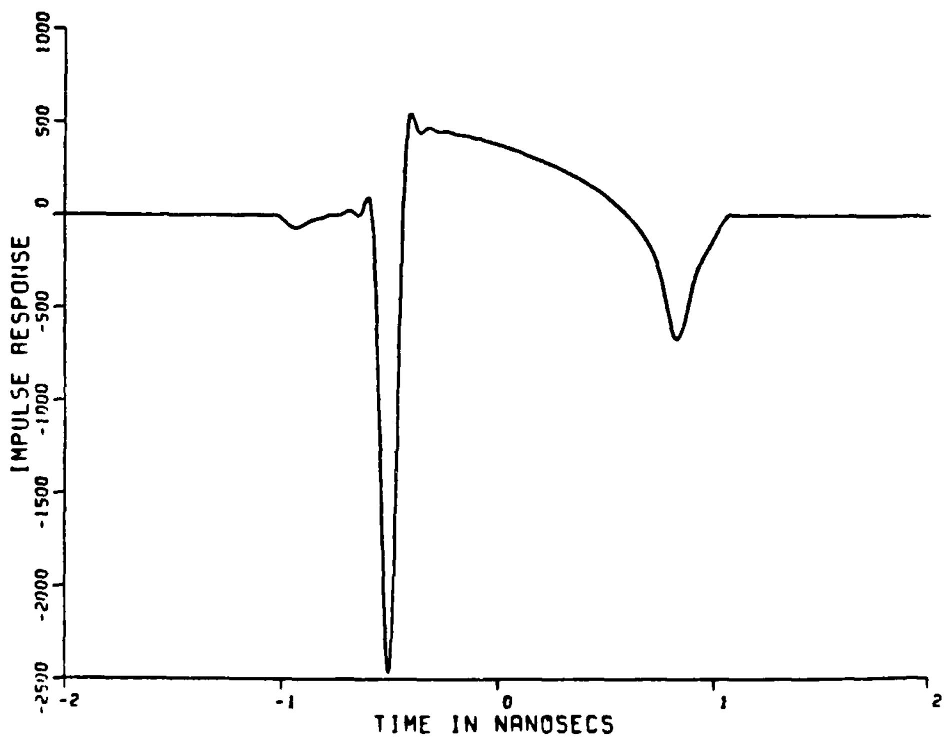


Figure 4-5. Similar description as Figure 4-3, except frequency samples are taken every 500 Mhz

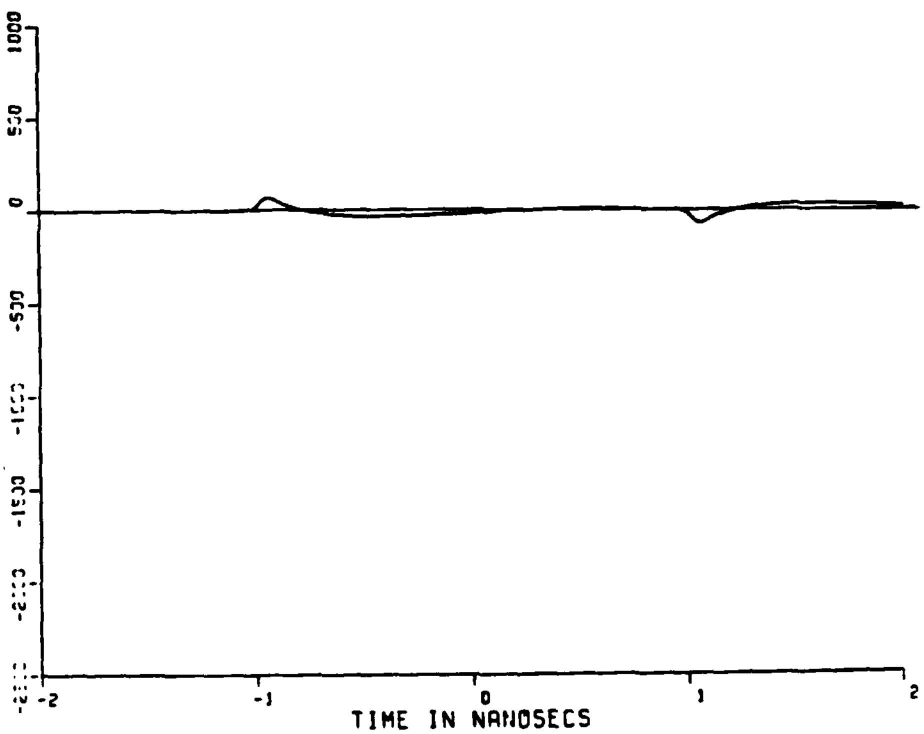


Figure 4-5a. Error of Figure 4-5 from Figure 4-2
(No magnification)

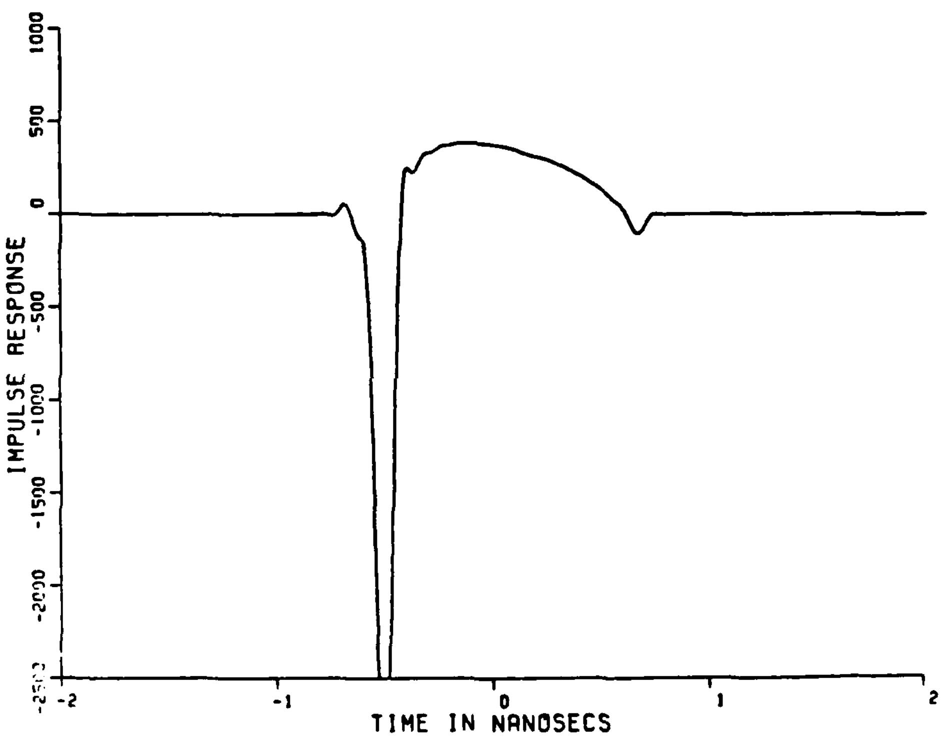


Figure 4-6. Similar description as Figure 4-3, except frequency samples are taken every 750 Mhz

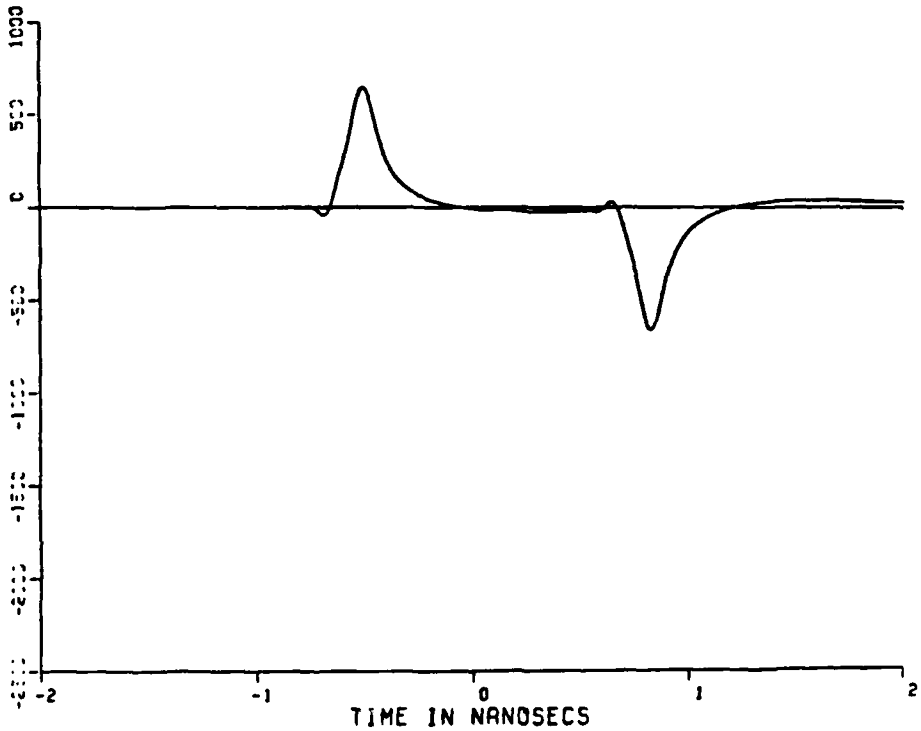


Figure 4-6a. Error of Figure 4-6 from Figure 4-2
(No magnification)

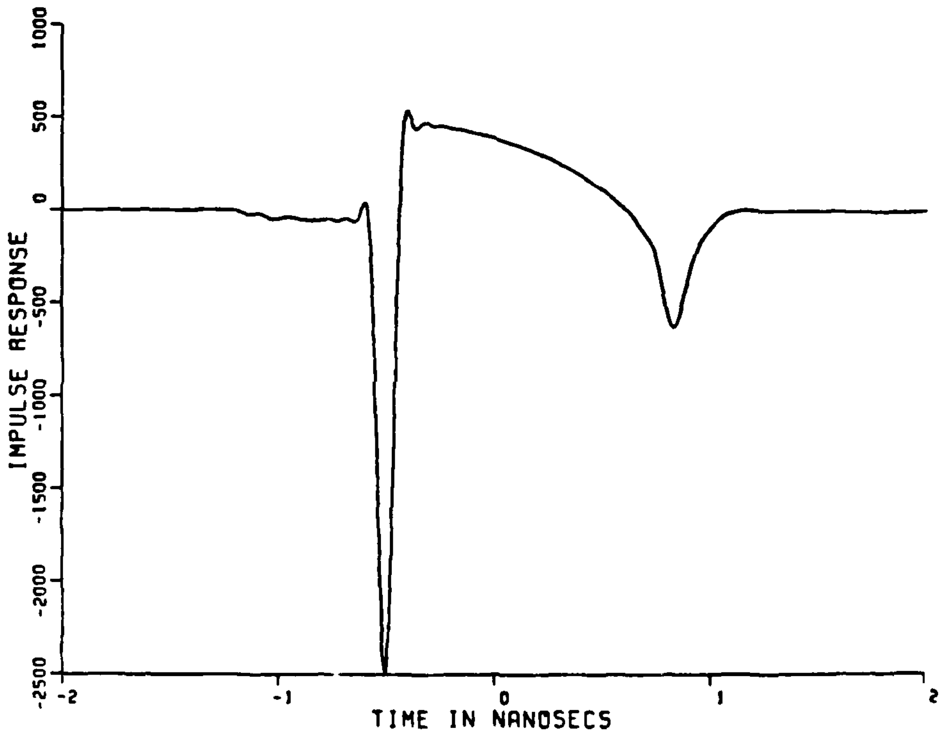


Figure 4-7. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from 2-D isotropic lattice samples taken with a safety factor of 1 over the range of 0 to 12 Ghz using Equation (4-3) and and (2-13)

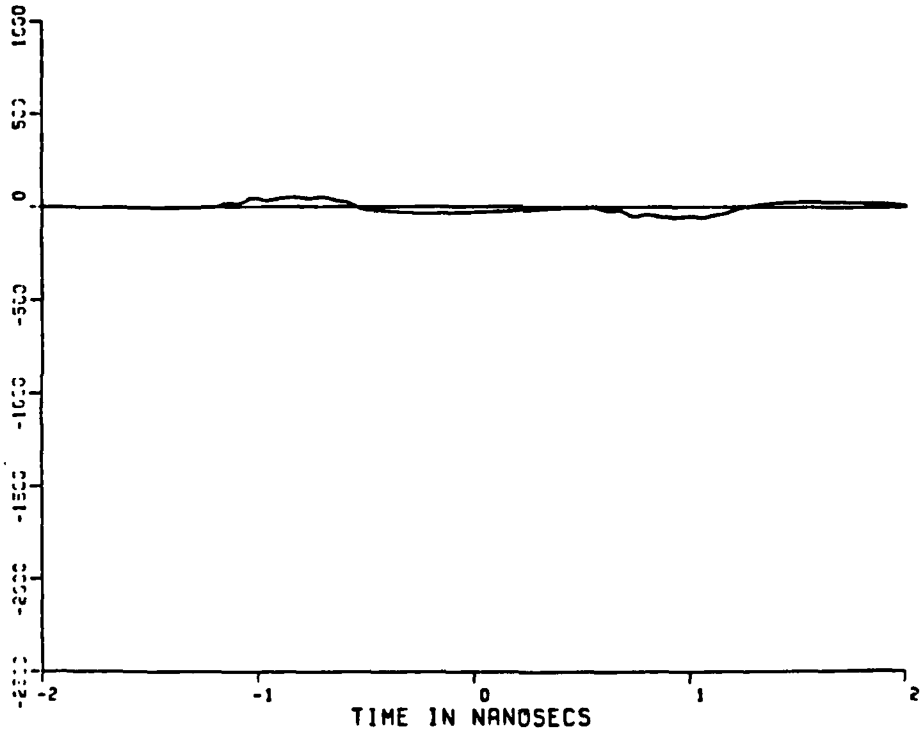


Figure 4-7a. Error of Figure 4-7 from Figure 4-2

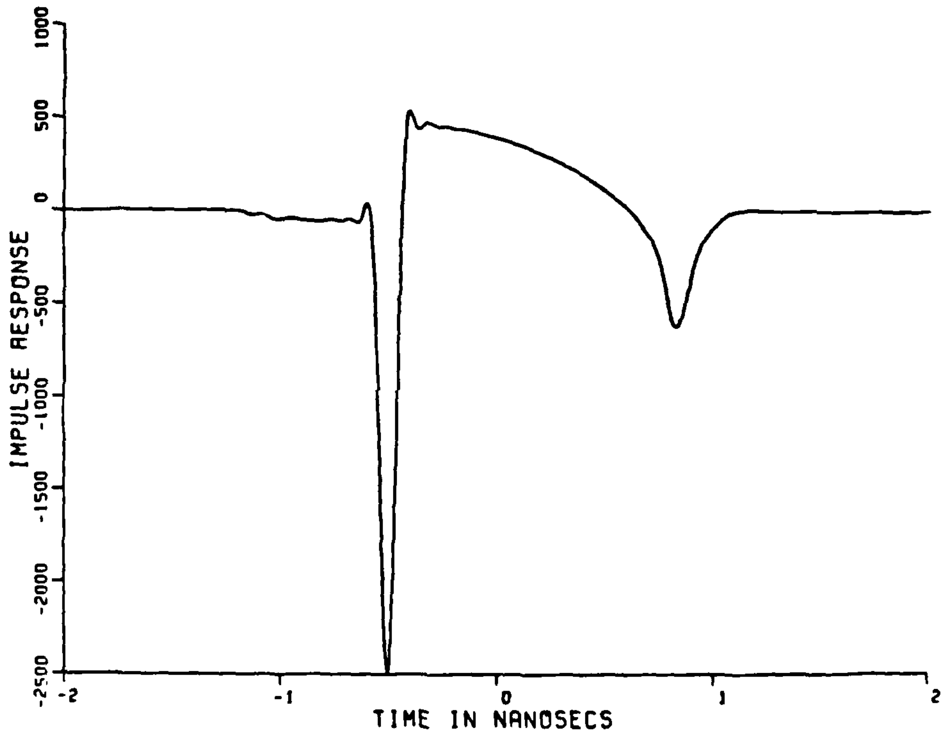


Figure 4-8. Similar description as Figure 4-7, except the aspect angle is .72°

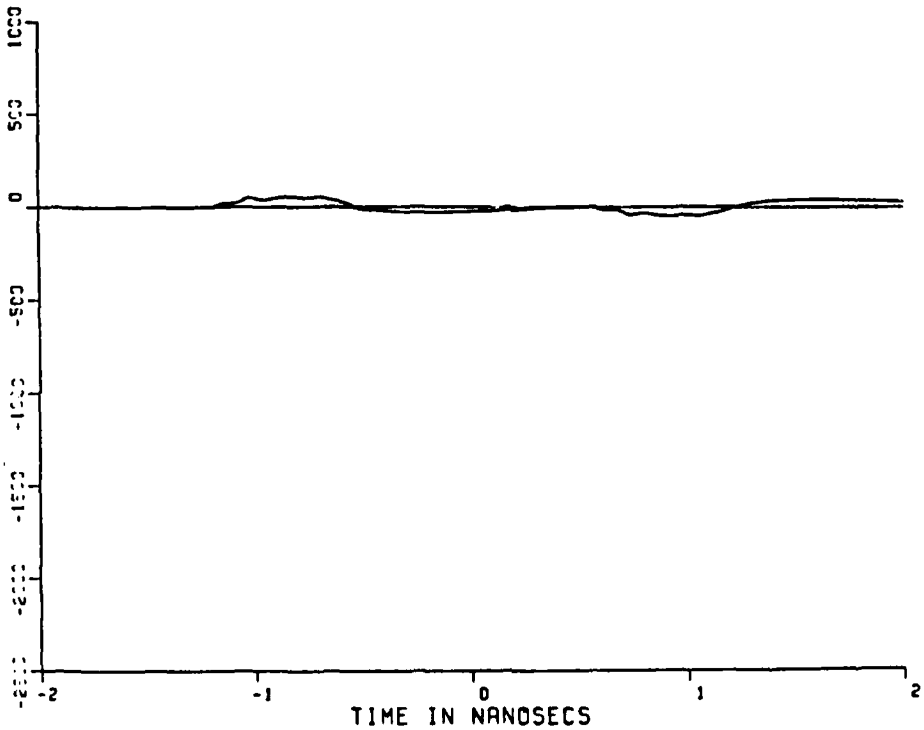


Figure 4-8a. Error of Figure 4-8 from Figure 4-2

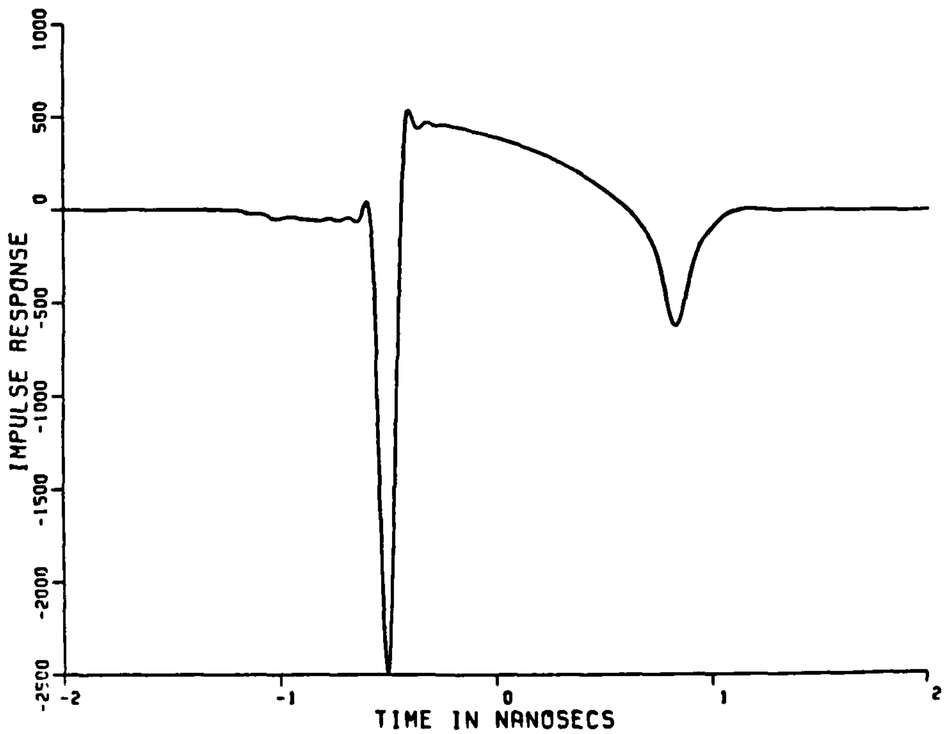


Figure 4-9. Similar description as Figure 4-7, except the aspect angle is 1.44°

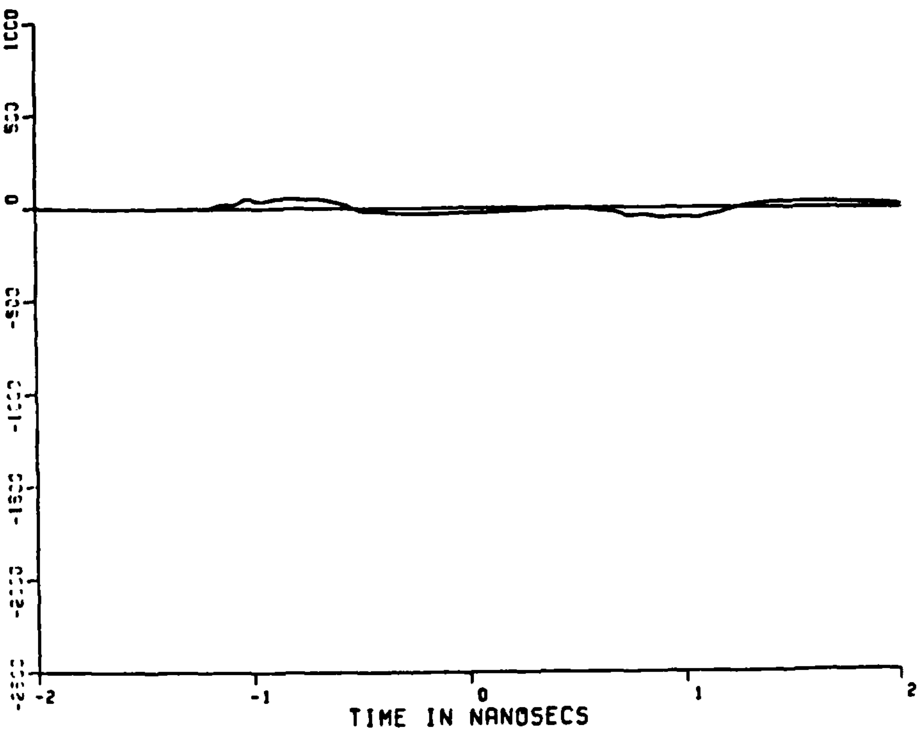


Figure 4-9a. Error of Figure 4-9 from Figure 4-2

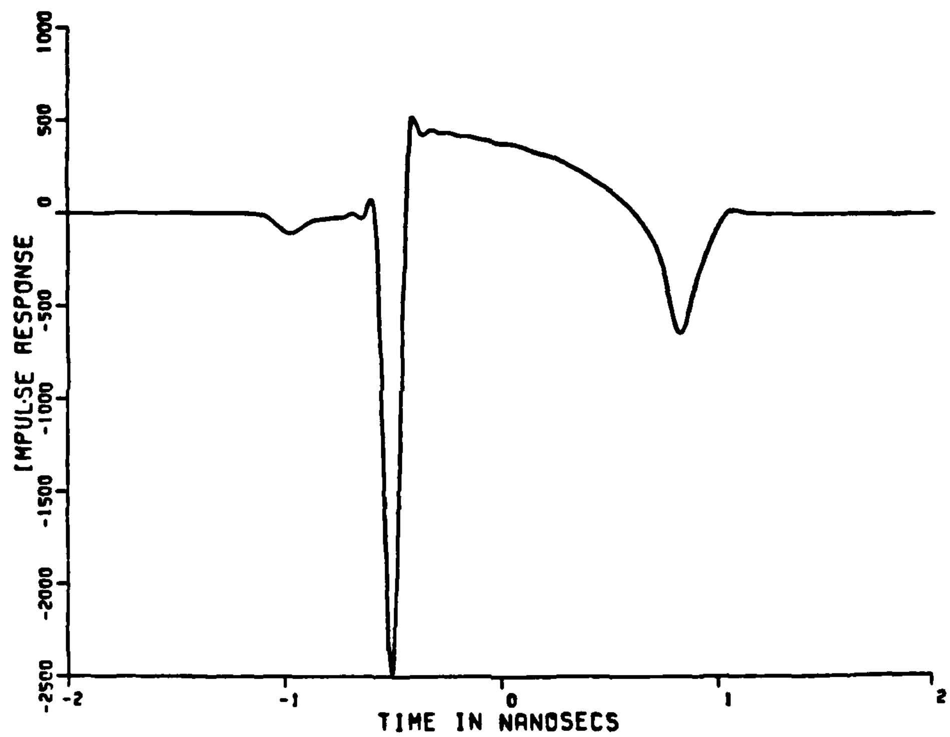


Figure 4-10. Similar description as Figure 4-7, except the aspect angle is 30°

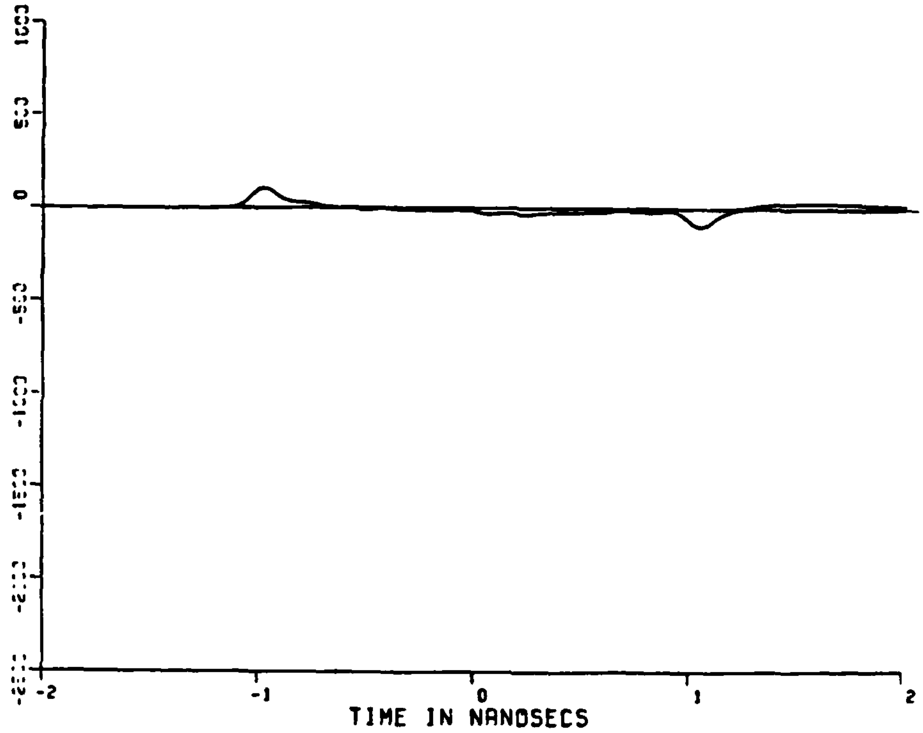


Figure 4-10a. Error of Figure 4-10 from Figure 4-2

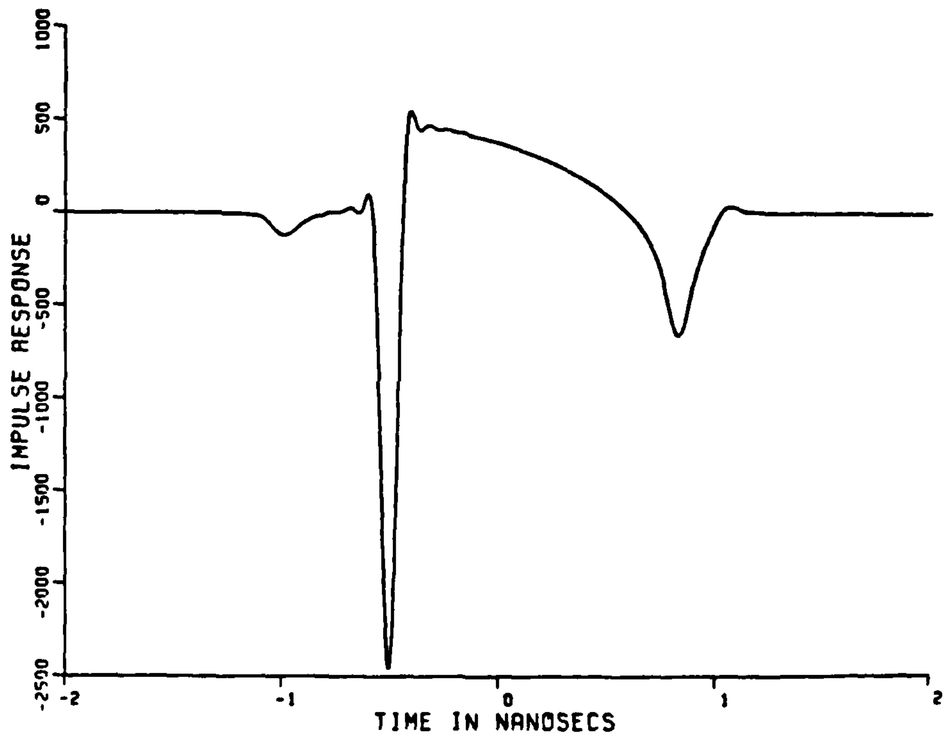


Figure 4-11. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from 2-D cubic lattice samples taken with a safety factor of 1 over the range of 0 to 12 Ghz using Equation (4-3) and (2-11)

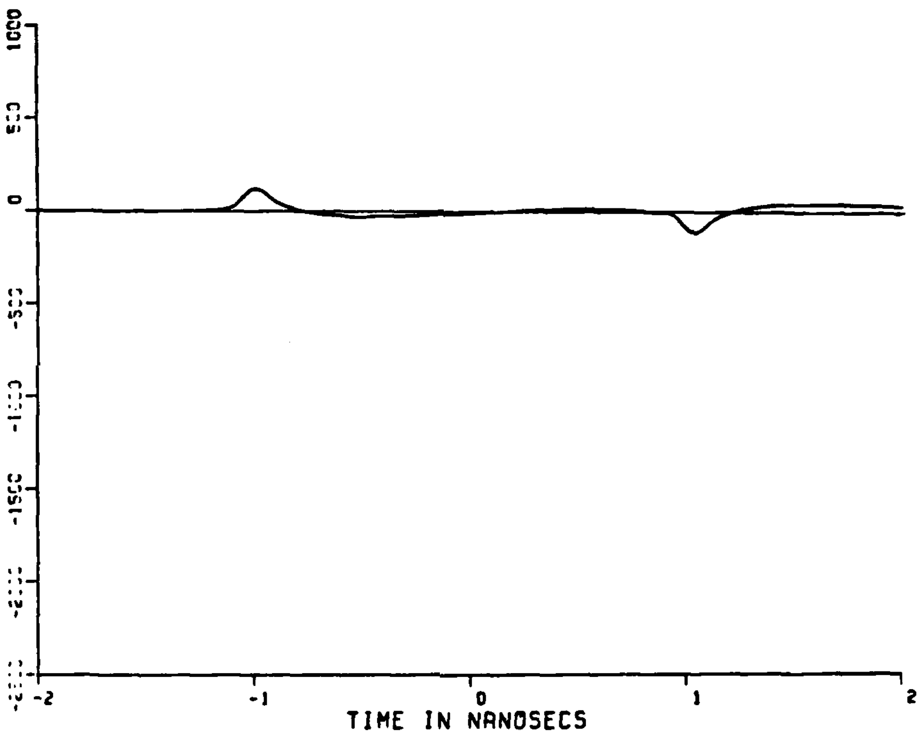


Figure 4-11a. Error of Figure 4-11 from Figure 4-2

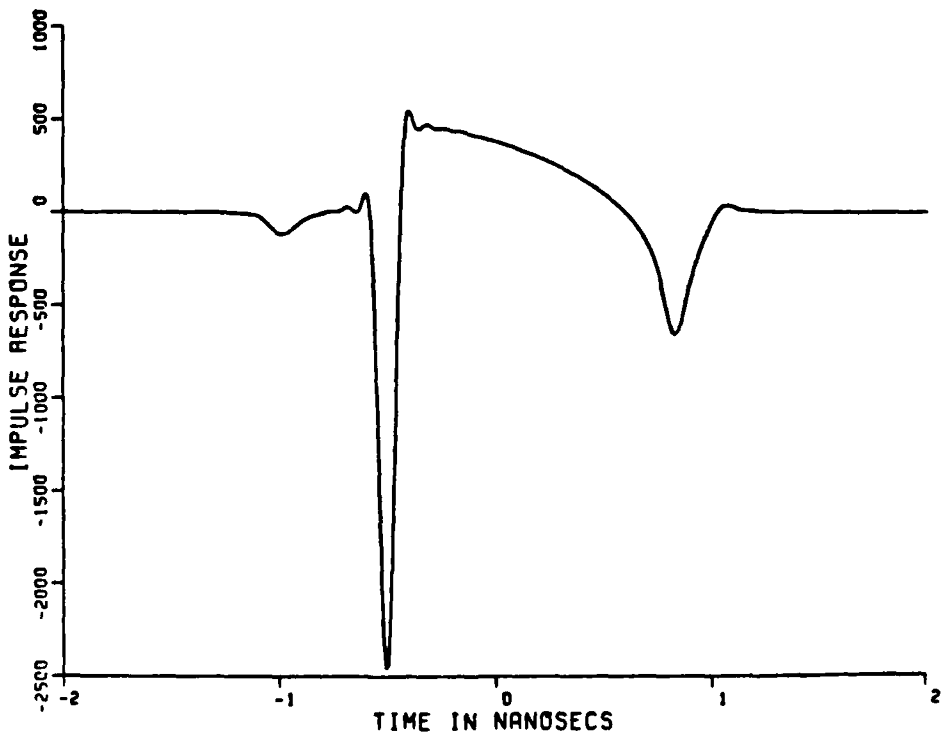


Figure 4-12. Similar description as Figure 4-11, except the aspect angle is 1.19°

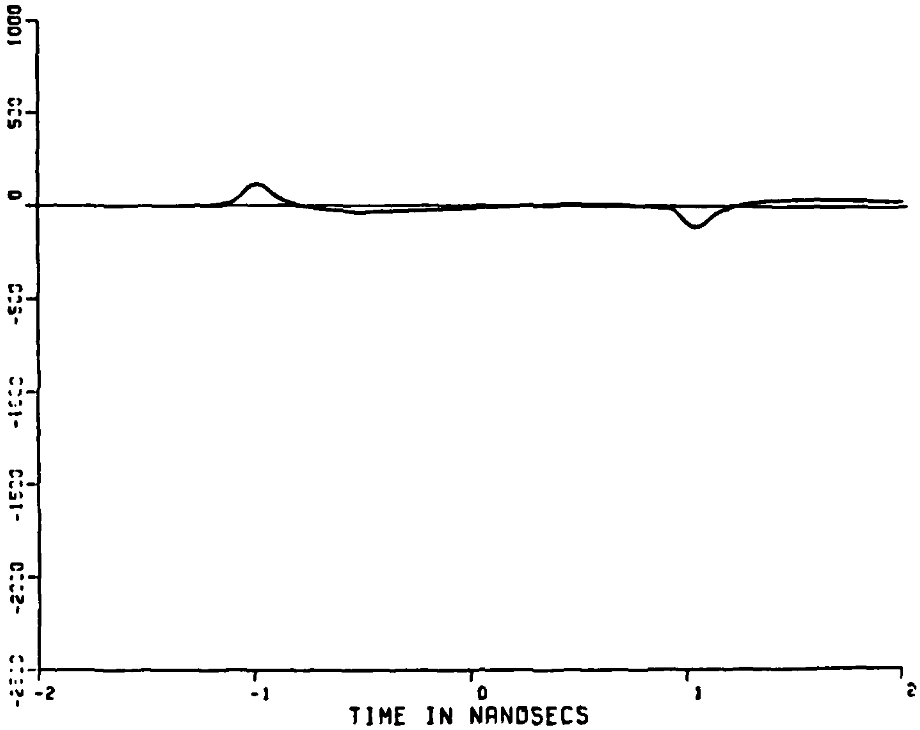


Figure 4-12a. Error of Figure 4-12 from Figure 4-2

Figure 4-13. Similar description as Figure 4-11, except the aspect angle is 2.39°

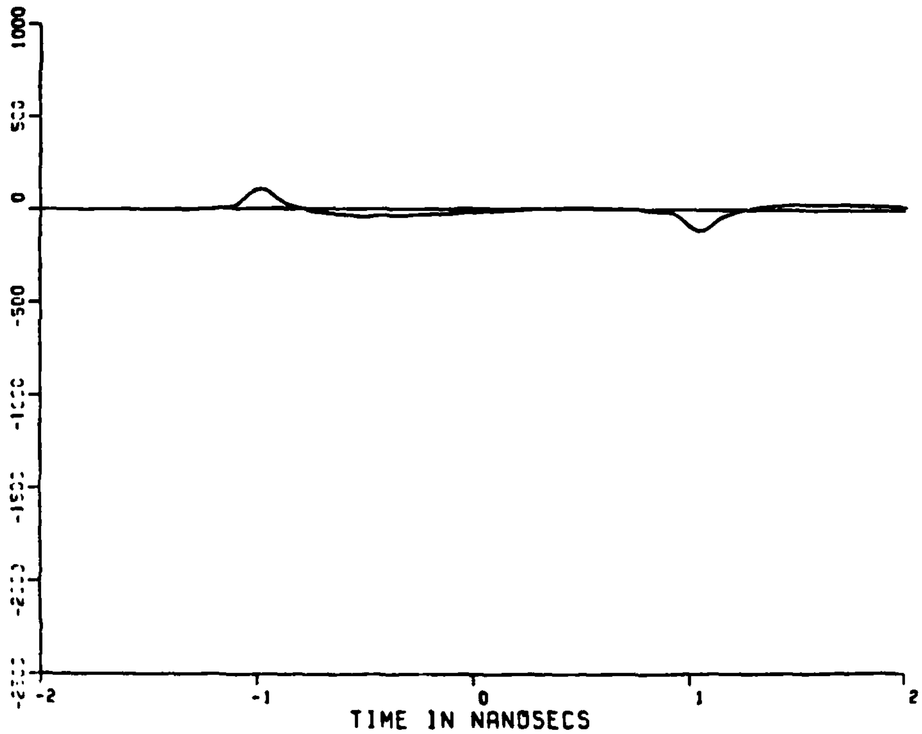


Figure 4-13a. Error of Figure 4-13 from Figure 4-2

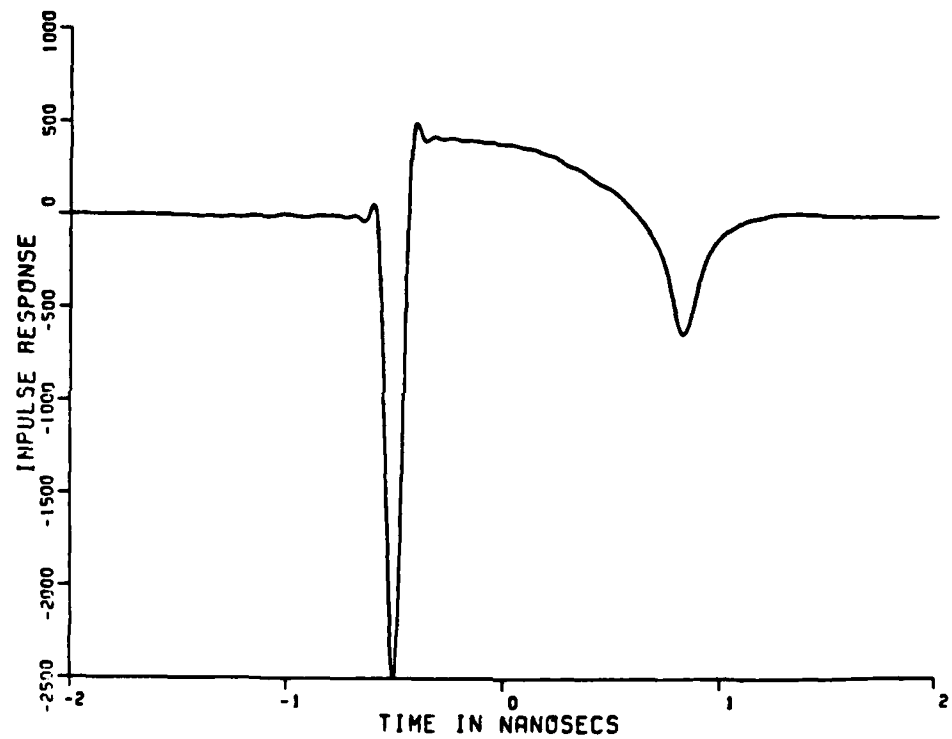


Figure 4-14. Similar description as Figure 4-11, except the aspect angle is 45°

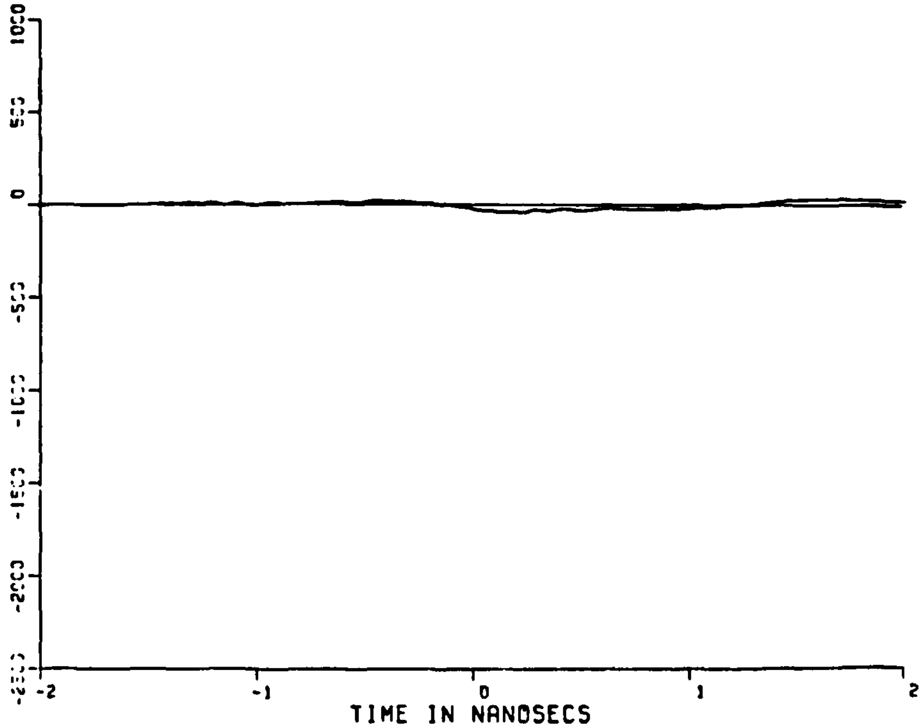


Figure 4-14a. Error of Figure 4-14 from Figure 4-2

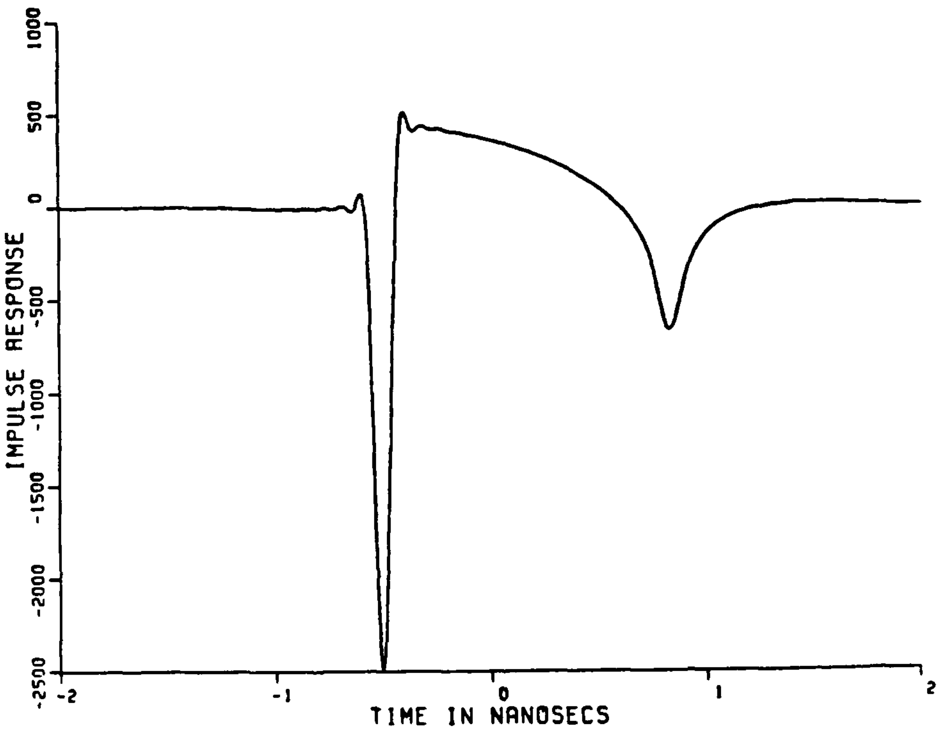


Figure 4-15. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from a 2-D isotropic lattice samples taken with a safety factor of 3 over the range of 0 to 12 Ghz using Equation (4-3) and (2-13)

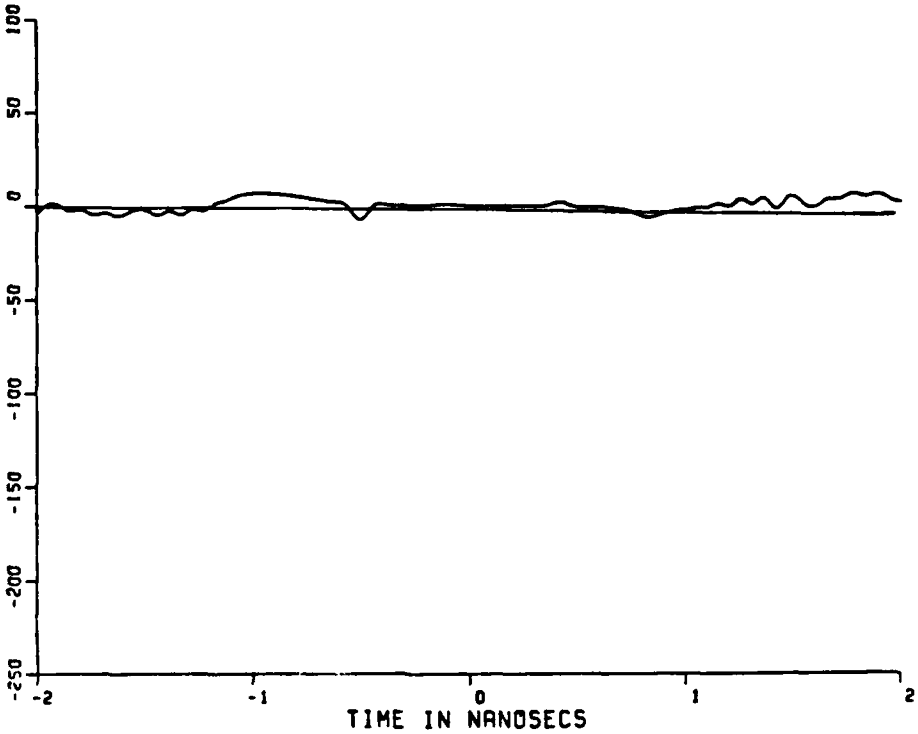


Figure 4-15a. Error of Figure 4-15 from Figure 4-2
(Magnification = 10x)

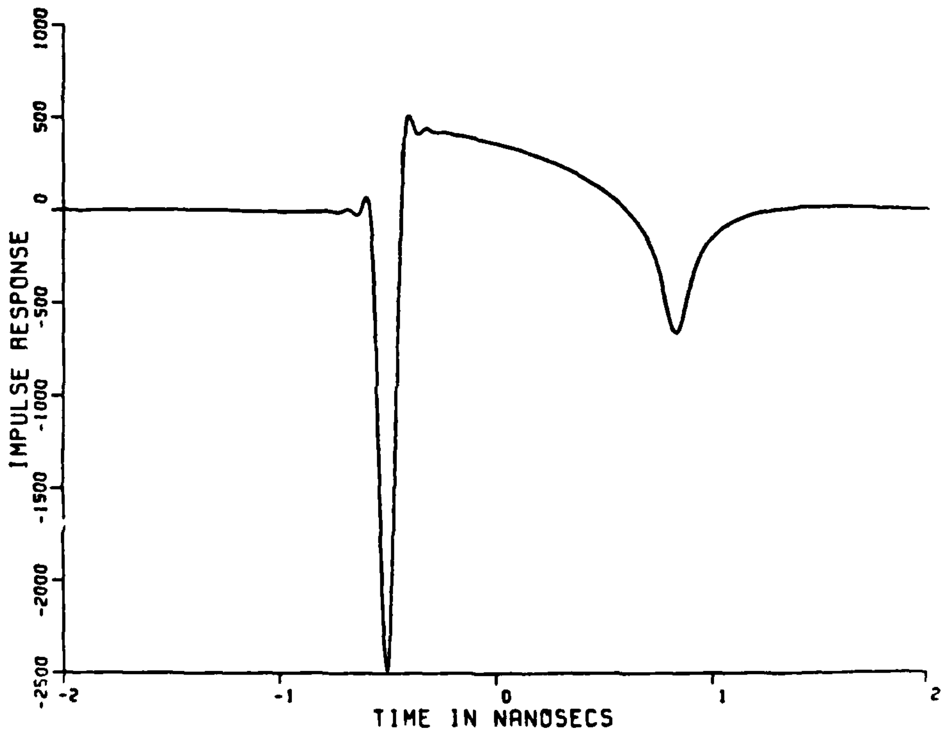


Figure 4-16. Similar description as Figure 4-15, except the aspect angle is 0.35°

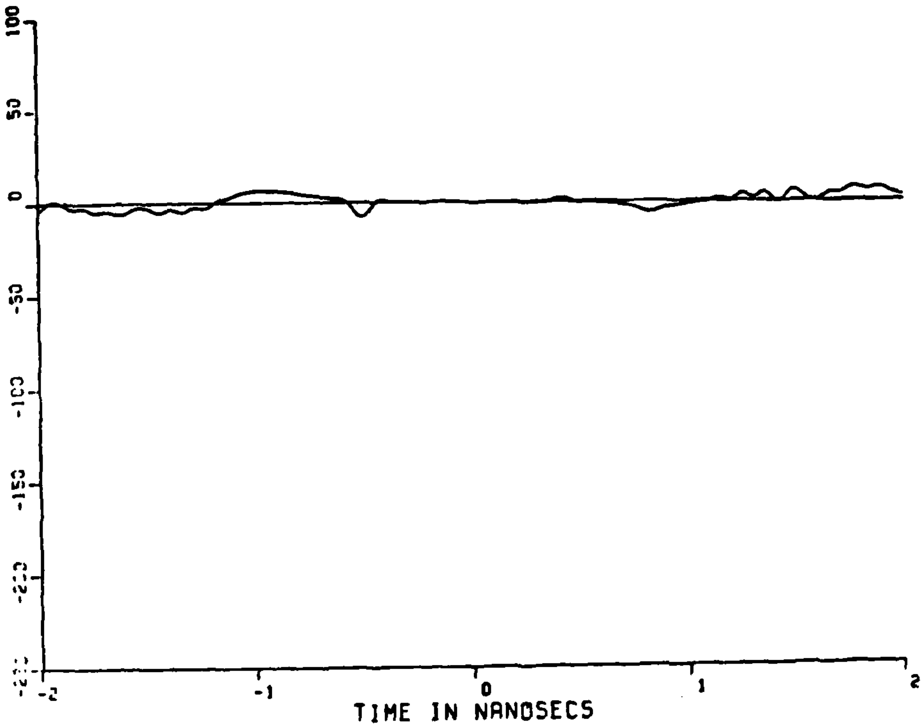


Figure 4-16a. Error of Figure 4-16 from Figure 4-2
(Magnification = 10x)

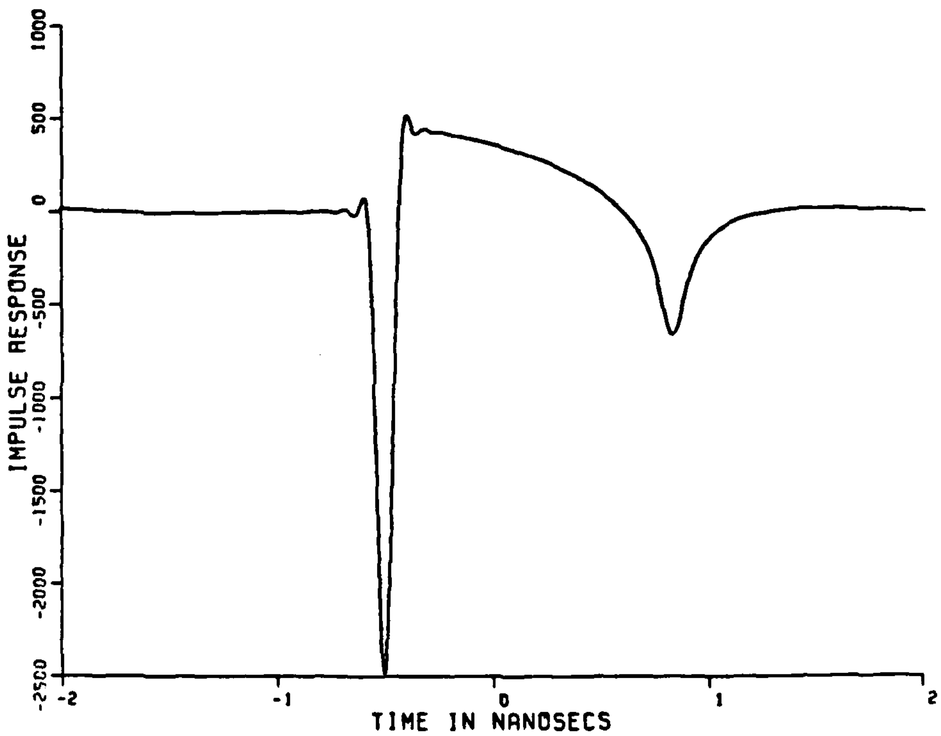


Figure 4-17. Similar description as Figure 4-15, except the aspect angle is 30°

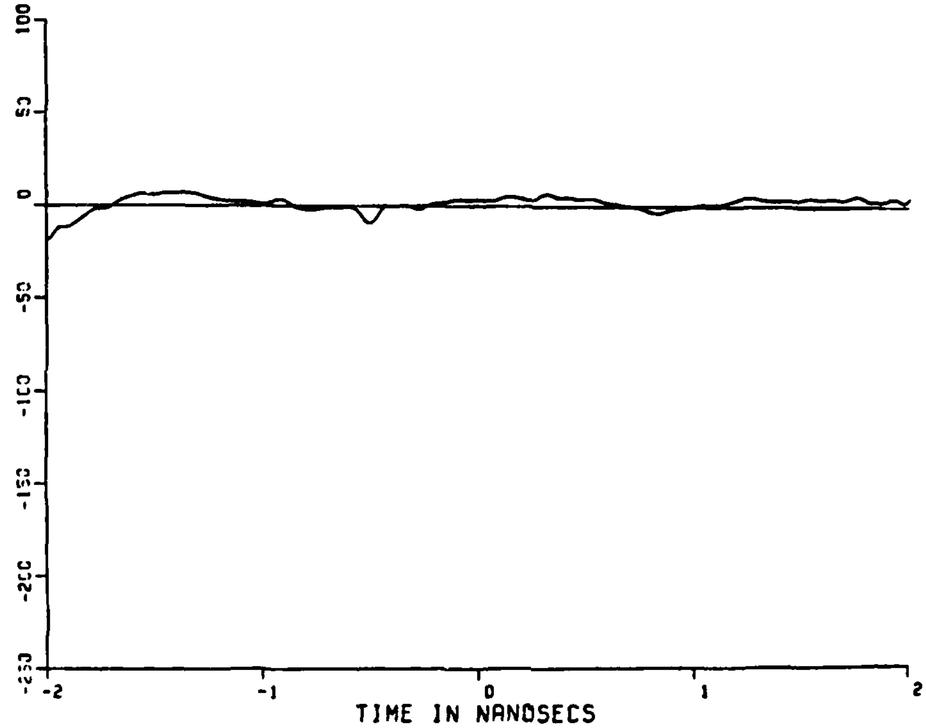


Figure 4-17a. Error of Figure 4-17 from Figure 4-2
(Magnification = 10x)

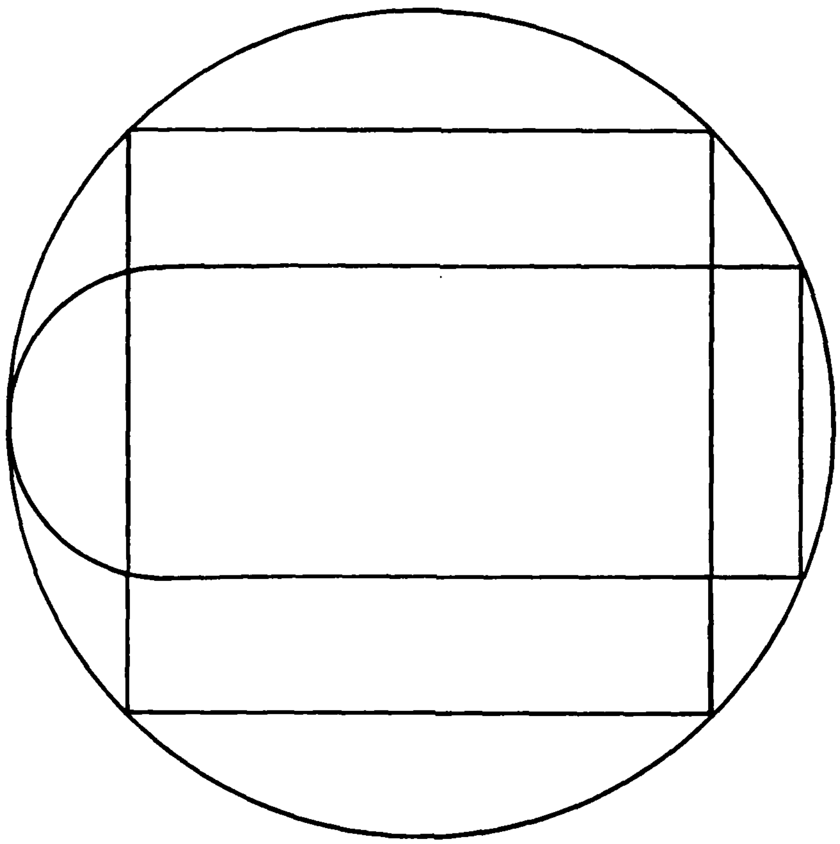


Figure 4-18. Dimensional perspective among the sphere, cube and sphere cap cylinder used in the example
(Diameter of sphere = 6 inches)

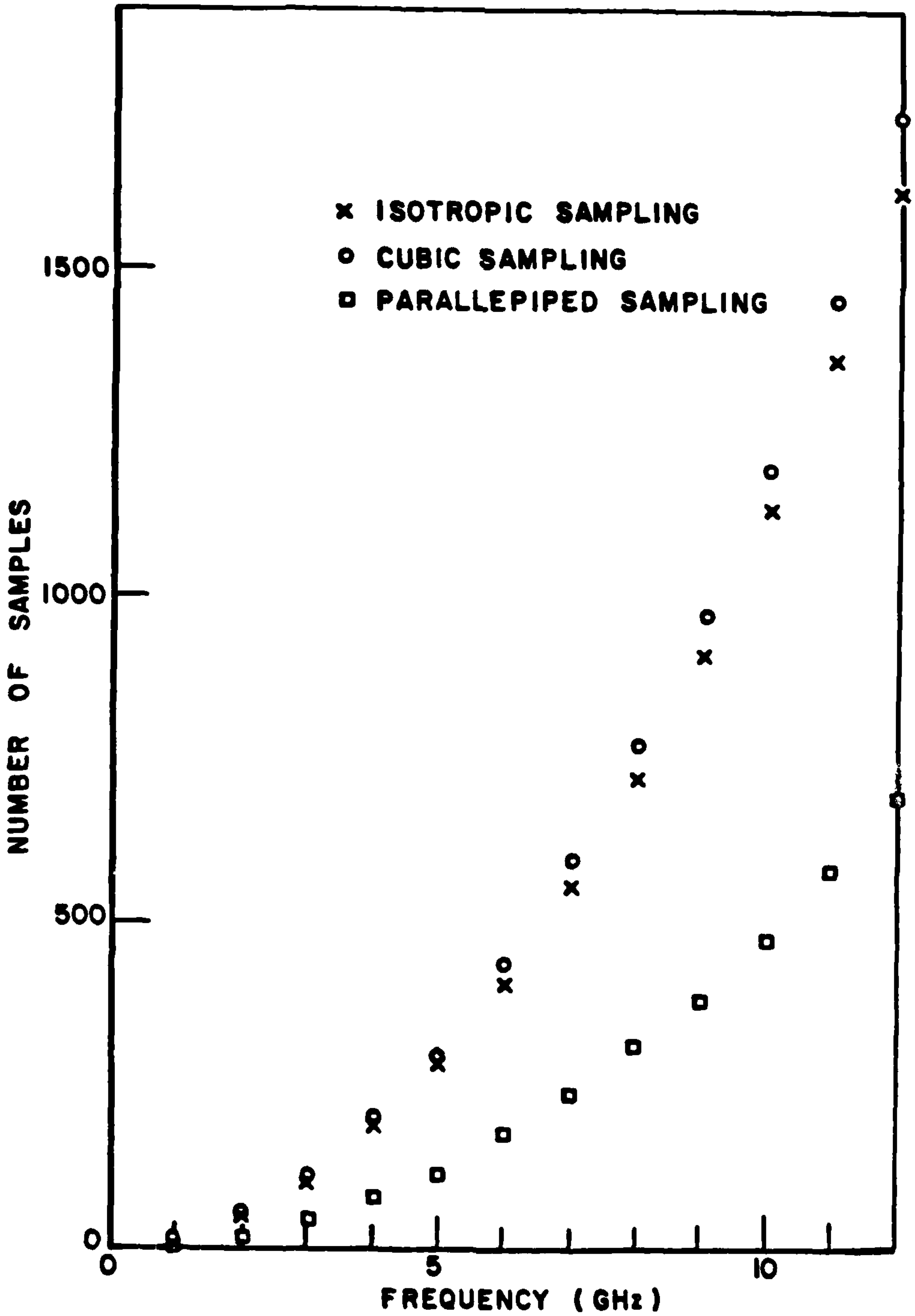


Figure 4-19. The number of samples specified by the different types of sampling lattices on a sphere cap cylinder described in Figure 4-18

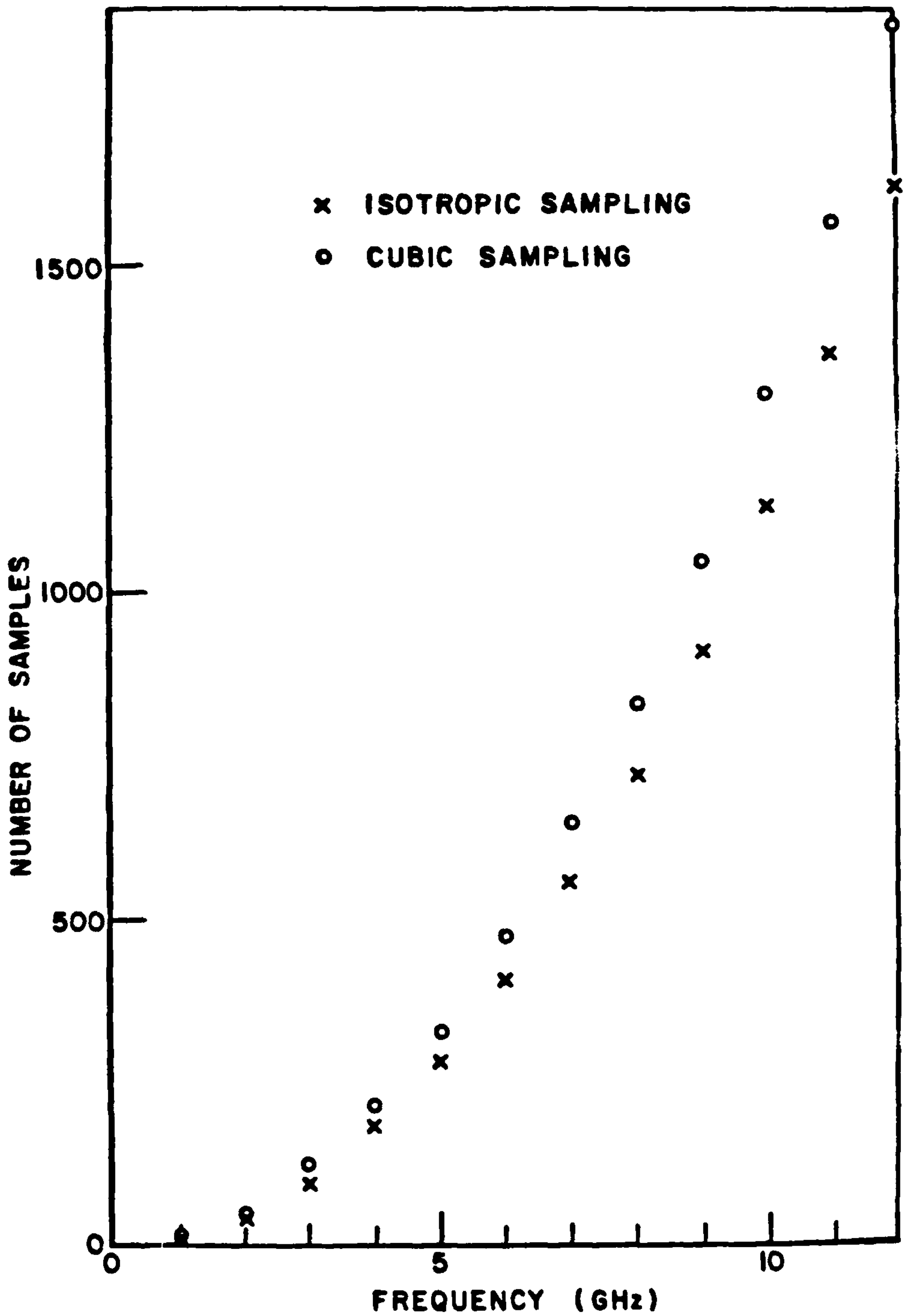


Figure 4-20. The number of samples specified by two types of sampling lattices on a 6" sphere

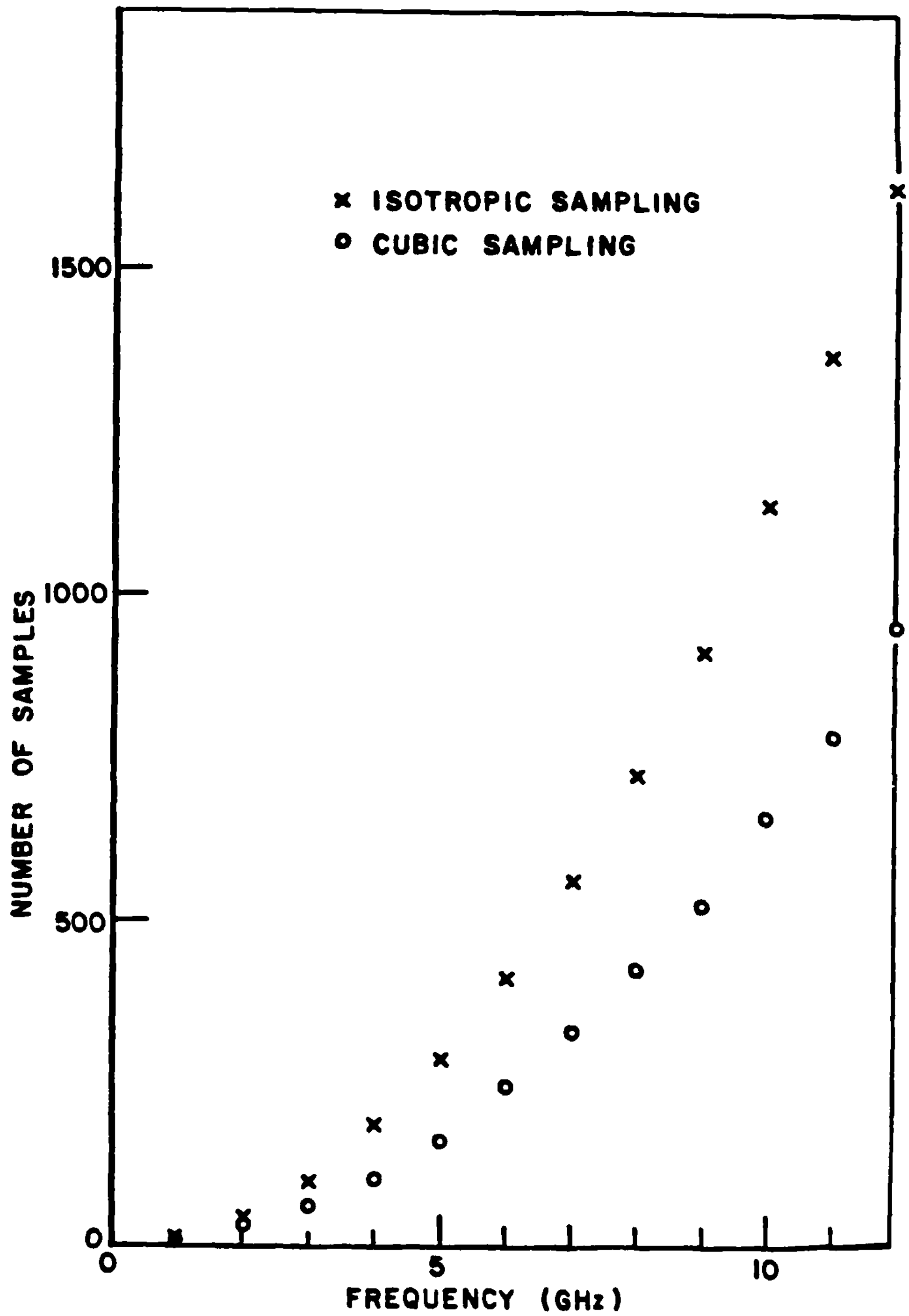


Figure 4-21. The number of samples specified by two types of sampling lattices on the cube described in Figure 4-18

CHAPTER V

TWO DIMENSIONAL IMPULSE RESPONSES

Thus far, the discussion has focused only on the number of samples required. The smaller the number of samples, the less measurement time is required. However, if the two dimensional impulse response is of interest, then the transform method must also be considered. This chapter will discuss the potential time consumption problem of multi-dimensional Fourier transform plus some possible solutions. Image reconstruction using the spatial impulse response is also considered.

While an isotropic enclosure may be more efficient to enclose a sphere than a cube, it is not as easy to do three dimensional discrete Fourier analyses as the cubic enclosure. Most of the discrete Fourier transform techniques are developed to fit equally spaced data. In another words, programs are written to perform readily on the cubic sampling lattice. Any other sampling grid data must be interpolated to the cubic grid either before or after the discrete Fourier transform step; otherwise, the proper representation cannot be achieved. Sometimes, interpolation may also be desired for the cubic sampling grid data, as in the case of higher resolution requirement than

measured. This type of interpolation requires a lot of computer time as pointed out in Mensa et al.'s paper [8].

Let's consider an example to estimate the time involved.

From Chapter IV:

1) Interpolation for 200 plotting points using 1626 isotropic samples is about 5 mins.
2) Interpolation for 200 plotting points using 1876 cubic samples is about 2.5 mins.
3) The number of samples required on a sphere cap cylinder for circular, rectangular and squared enclosures are 1626, 696, 1740 respectively. (Table 4-2)

Compact range measurement facility at O. S. U. per Walton [12]:

The measurement system response time is about:

1) 0.2s/data point, if frequency scan is used.
2) 1 min/frequency ring, if angular scan is used.

The time estimation for each type of the sampling measurement on the sphere cap cylinder and two dimensional interpolation is presented in Table 5-1. (For the convenience of the reader, all tables and figures of Chapter V are grouped together at the end of the chapter.) The interpolation time for the rectangular enclosure data may also be considered zero. By remembering a scale factor, the data can be processed as in a squared grid. If the proper signal representation or a finer resolution is required, the interpolation step is still

unavoidable. Therefore, the numbers in the table do give a fair comparison, as all data are brought to the same level - squared grid representation. The interpolation step plays an important role on the decision of which type of sampling to use. The saving in measurement time is sometimes balanced out by the data processing time.

The Nyquist criteria (Equations (2-19), (2-20)) plus the polar transformation (Equation (2-15)) described earlier give new perspective for the efficient but non-cubic enclosure. Now one only needs to interpolate for a sufficient number of frequency rings. This work is done by program INTERPOL(see Appendix B). Each frequency ring is transformed to the spatial domain individually and summed together to get the total time response. The former is the work of program INTEGFFT (see Appendix B); the latter is the job of program SUM3D (see Appendix B). The interpolation performed this way may require as much time as the interpolation onto the squared grid. It nevertheless gives an alternative way to the solution of the problem. Now, another question may be raised: Why perform interpolation if it is so time consuming?

The interpolation may be avoided, if the two dimensional time response is the only interest. All one has to do is to measure the frequency rings and then process the data as described before. However, if one desires the impulse response of a target at a particular aspect angle or the response over a frequency ring other than those measured, one is required to develop a different interpolation scheme than the one described in the theory section. A possible way to reduce the

interpolation time is to derive a faster interpolation scheme or a general multi-dimensional Fourier transform technique which operates on any grid. A parallel processor which uses optics may be another possibility in terms of hardware. A lens system has a Fourier transform relationship between the source and the image [13].

The Mie solution for a sphere is a very good data example; except its backscattered response is isotropic. A proper choice of test object must have non-isotropic property. The sphere shifted off the centre of the plane is one possibility, but it only has variation in the phase term and not in the magnitude term. Since most of the other exact solutions are not readily available, a first order UTD solution for a finite circular cylinder [14] is used; with the caution that UTD gives a valid approximation to the exact solution at high frequency.

The size of the circular cylinder is chosen to be six inches in length and three inches in diameter. Having chosen the size of the cylinder, one has defined the low frequency limit of this UTD model to about 2 Ghz. Since the step and ramp response of an object reduce the need for the high frequency information, only the impulse response of this cylinder solution is considered here.

Figure 5-1 is the two dimensional impulse response (Equation (2-9)) for the above circular cylinder. The frequency samples are first generated on the grid points defined by Equation (4-5) over only 180°. In this case, the spatial impulse response is assumed to settle after the wave has passed over four times the length of the object at every aspect angle. The rectangle so designated has dimensions: 24" in

length and 12" in diameter. The periodic lattice is chosen as in Figure 2-3. Consequently,

$$\bar{v}_1 = 24" \begin{bmatrix}1\\0\end{bmatrix} \qquad \bar{v}_2 = 12" \begin{bmatrix}0\\1\end{bmatrix}$$

The guard band is 9" on each side of the cap of the cylinder and 4.5" on the circular surface. Then,

$$\bar{u}_1 = \frac{2\pi}{24"} \begin{bmatrix}1\\0\end{bmatrix} \qquad \bar{u}_2 = \frac{2\pi}{12"} \begin{bmatrix}0\\1\end{bmatrix}$$

Since,

$$l_1\bar{u}_1 = l_1|\bar{u}_1|\hat{k}_1$$

$$l_2\bar{u}_2 = l_2|\bar{u}_2|\hat{k}_2$$

where

$$l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots$$

Therefore, the sampling lattice is defined by:

$$l_1\bar{u}_1 + l_2\bar{u}_1 = l_1|\bar{u}_1|\hat{k}_1 + l_2|\bar{u}_2|\hat{k}_2$$

<=> points on the k-plane:

$$k_1 = l_1|\bar{u}_1| \text{ and } k_2 = l_2|\bar{u}_2|$$

where

$$l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots$$

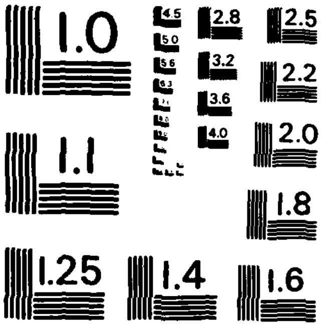
Then the sampled data are Fourier transformed discretely into the spatial domain. The discrete two dimensional Fourier transform is performed by an IBM subroutine HARM (see Appendix B). The frequency range used is 2 to 5 Ghz. The safety factor of one is used. The vertical polarization data are taken from the UTD solution on a major axis cut. Figure 5-1a is the contour plot of Figure 5-1. Figure 5-2 is also a two dimensional impulse response for a cylinder solution that has the same previous description. The differences are the technique of the Fourier transformation and the sampling locations in the frequency plane. It employs Mensa et al.'s method on 6 different frequency rings: 2.5, 3, 3.5, 4, 4.5, 5 Ghz. (Equation (2-18)) The samples are taken so that Equations (2-19) and (2-20) are satisfied. Again the samples are taken over 180°. Figure 5-2a is the contour plot of Figure 5-2.

Comparing Figures 5-1 and 5-2, one can see many similarities. The differences can be deduced by recalling their respective generating methods. Figure 5-1 has information scattered all over the frequency range; while, Figure 5-2 has vaules only over those frequency rings mentioned before. There is also the processing error involved.

One interesting thing is to be noted in Figure 5-1, or 5-2. If one records just the highest points within its neighborhood, one can trace out a rectangle. This may be easier to see on a contour plot (Figure 5-1a, 5-2a). This looks like one cross-section of the target. If 2π

AD-A162 553 SPACE-FREQUENCY SAMPLING CRITERIA FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT(U) OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F Y FOK AUG 85 2/3

UNCLASSIFIED ESL-714190-11 N00014-82-K-0037 F/G 20/14 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

solid angle information is available, then an image of the target can indeed be produced. Of course, the resolution is still governed by the highest frequency used. In this case the bandwidth used is quite narrow; hence, the resulting resolution is not high.

Figure 5-3 is generated similarly as Figure 5-2. It is generated using Mensa et al.'s method on 16 different frequency rings: 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10 Ghz. Figure 5-3a is the contour plot of Figure 5-3. The dimensions of the rectangle defined by the high peaks do correspond to the major axis cross-section of the circular cylinder described earlier. One may wonder how a rectangle is concluded from Figure 5-3a. There are a few clues available. The T_i's are quite obvious from the high peaks on the illuminated side. The final time (T_f) of the one dimensional impulse response, as one recalls, is defined by the beginning of the exponential decay of the signal. The final peaks at the coordinates (5,3) and (5,-3) are indeed higher than any point $x_1 > 5$. The ridges and valleys on the shadow side ($x_1 > 5$) of the cylinder is fairly straight.

Let's consider the case where this imaging theory converges to Lewis-Bojarski's work. Figure 5-4 is the same as Figure 5-3 except data are taken over 360°, or illuminated in every direction on the two dimensional plane. Figure 5-4a is the contour plot of Figure 5-4. As expected, the peaks (or T_i's) in the figure trace out a rectangle which is the major axis cut of the circular cylinder. One may note that literature today usually presents data plots using the absolute value of the amplitude. One must be careful when confronted by these plots.

As Figures 5-5 and 5-5a have shown, the absolute value of the amplitude does not usually tell the whole picture. Figure 5-5 plots the absolute value of the amplitude in Figure 5-4. Figure 5-5a is the contour plot of Figure 5-5. The major axis cross-section is no longer as well defined by the peaks as in Figure 5-4 or 5-4a. Nevertheless, producing an image using the spatial impulse response is very promising.

Although this thesis' imaging theory is not as 'rigorous' as Lewis and Bojarski's work, the theory is more flexible in application. The object is only required to be illuminated at the aspect angles over a 2π solid angle. The shadowed side information is also employed in the image reconstruction process. There is no need for any assumption on the object's shadowed side geometry. If the start and the end of the object's one dimensional impulse response for every aspect angle over half of the 4π solid angle are well defined by T_i's (illuminated) and T_f's (shadowed) as described earlier, then an image of the object may be produced using the T_i's and T_f's. For T_i's and T_f's not defined distinctly, careful interpretation on the spatial impulse response must be used. In the case of a full 4π illumination, the image may be reconstructed using only T_i's information. This converges to Lewis-Bojarski's identity. This theory is better in utilizing data information available but it lacks a concrete proof.

Let's investigate this imaging possibility further using a six inch diameter metallic sphere whose centre is located three inches off the centre of the reference plane on the x_1-axis. The cubic sampling

lattice used has a safety factor of three. The frequency range used is 0 to 12 Ghz. Again, the frequency data are taken out of the Mie solution. The sampling lattice is defined by Equation (4-5) over half of the plane ($\pi<\theta<2\pi$). The data are weighted by the two dimensional cosine tapering function. The two dimensional version of Equation (4-2) is replacing the variable n by the radial distance from the centre of the k-space. The shape of the function is Figure 4-1 rotated around the weighting axis. The weighted data are Fourier transformed into the spatial domain and presented as Figure 5-6. Figure 5-6a is the contour plot of Figure 5-6.

Again the initial speculars trace the illuminated side of the sphere nicely. The T_f's are not as well defined as the finite cylinder case. More interpretation work is required. The valley on the shadow side shows a curvature. This may be indicating the back side of the object having a curvature. The radius of the valley's curvature is about 3π inches, which is the equivalent distance a wave would creep before scattering back in the transmitting direction (Figure 5-7). The radius of the curvature created by the initial speculars is about six inches, which is the 2R distance travelled by the wave. The first R is the distance travelled to the target. The second R is the distance travelled by the wave scattering back from the target. Using this information, the circle with the three inch radius can be formed.

Of course, if one uses data over the full plane as Lewis-Bojarski's work, then there is no need for the previously described type of pattern recognition interpretation. The initial speculars usually define the perimeter of the object, if it is a smooth convex body. The perimeter indicated is only one cross-section of the target on the major axis. If more cross-sectional information of the target is available, then an image of the whole object may be produced. The potential on the image reconstruction is high, but more research work is required in the area; particularly in the pattern recognition area.

TABLE 5-1

AN EXAMPLE: TIME ESTIMATION

Type of enclosure	Measurement time (mins)	Interpolation time (mins)	Total time (mins)
circular	5.42	40.09	45.51
rectangular	2.32	8.47	10.79
squared	5.80	-	5.80

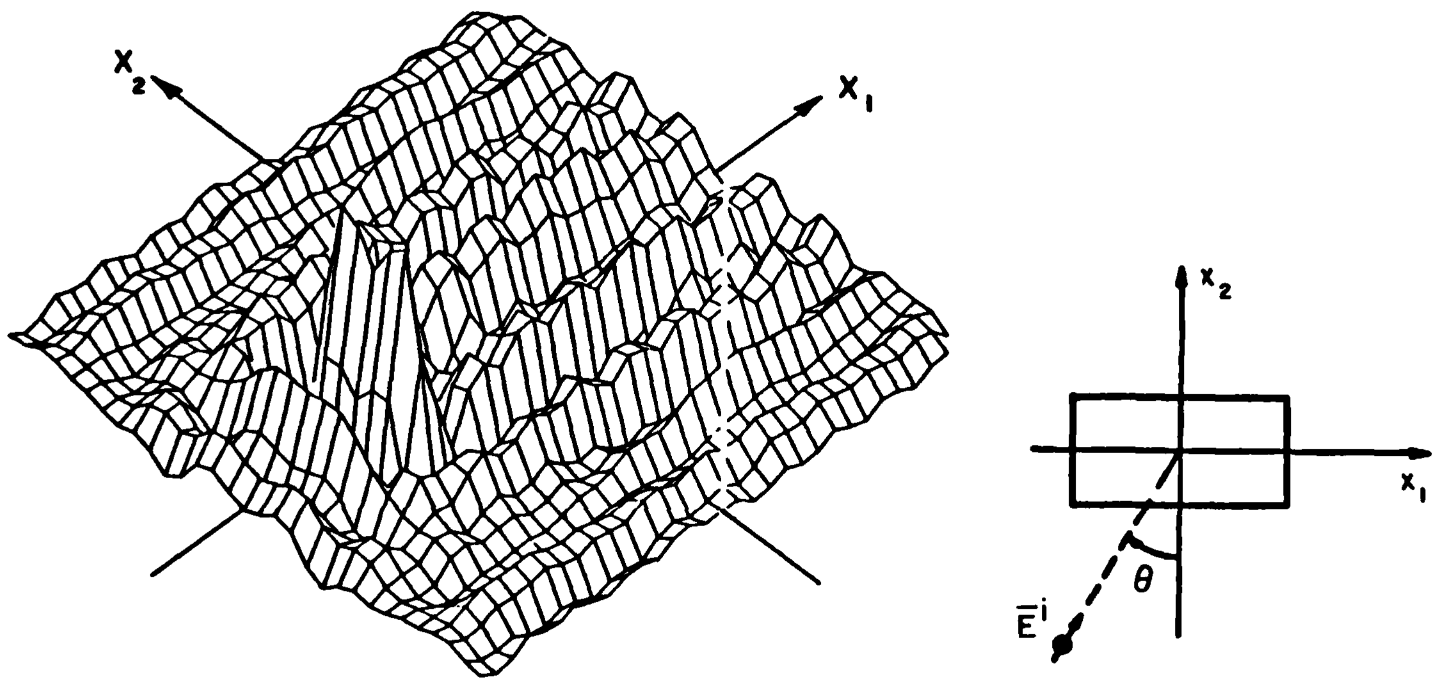


Figure 5-1. 2-D impulse response of a finite metallic circular cylinder with 6" in length and 3" in diameter. Frequency samples are taken at the cubic sampling lattice over the range of 2 to 5 Ghz and Fourier transformed discretely into the spatial domain. Data are taken over the half plane: $0 \leq \theta \leq \pi$

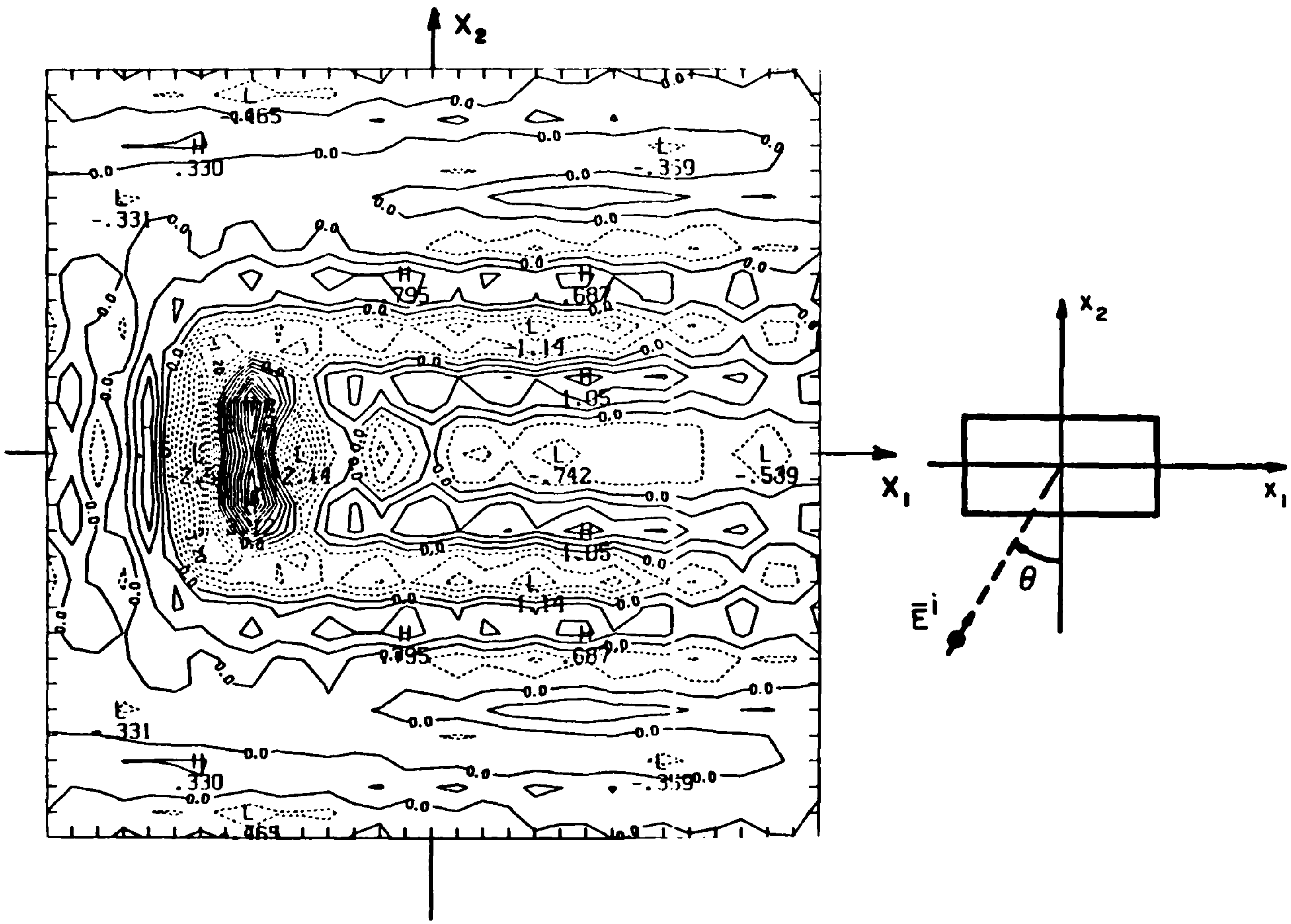


Figure 5-1a. Contour plot of Figure 5-1

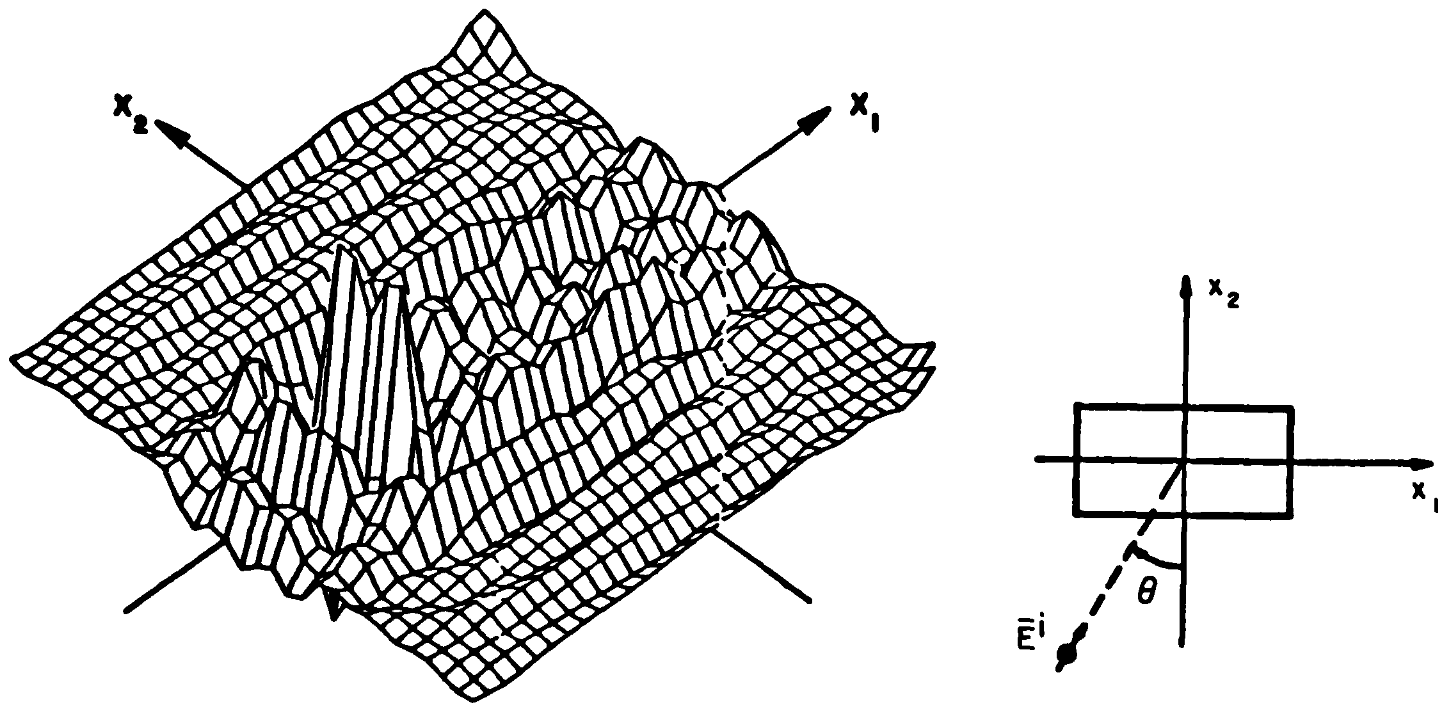


Figure 5-2. 2-D impulse response of a finite metallic circular cylinder with 6" in length and 3" in diameter. Frequency samples are taken over 6 rings: 2.5, 3, 3.5, 4, 4.5, 5 Ghz and transformed into the spatial domain via Mensa et al.'s method. Data are taken over the half plane: $0 < \theta < \pi$

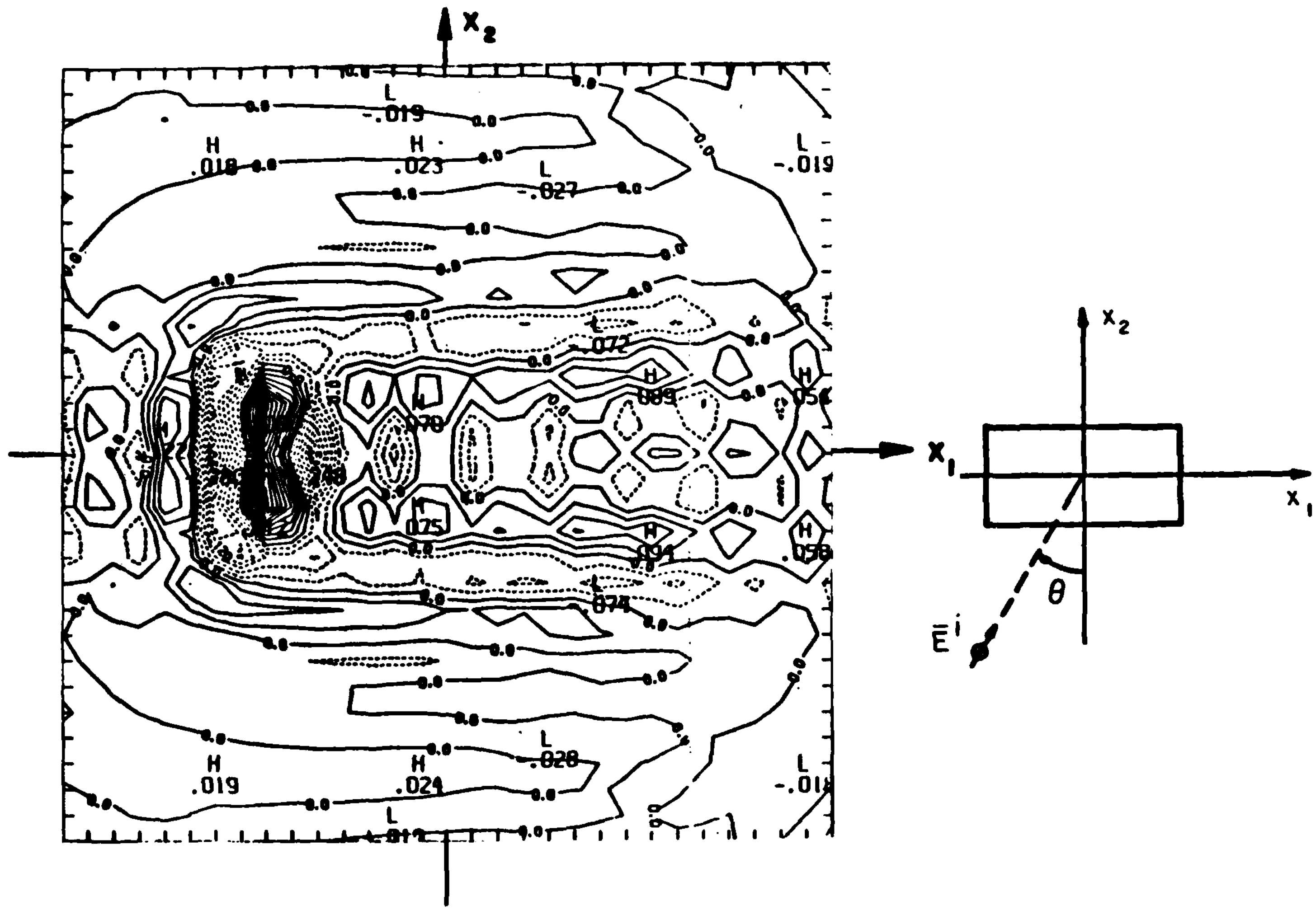


Figure 5-2a. Contour plot of Figure 5-2

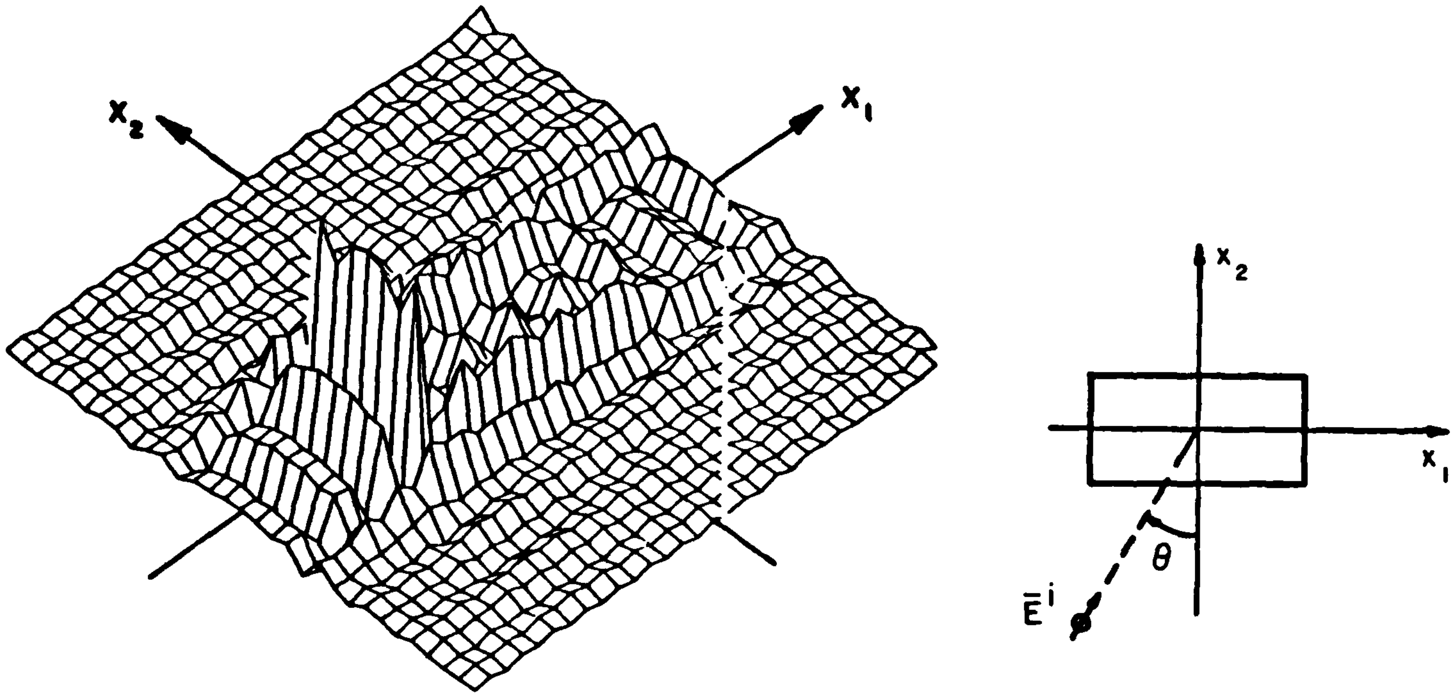


Figure 5-3. Similar description as Figure 5-2 with 10 additional frequency rings: 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10 Ghz

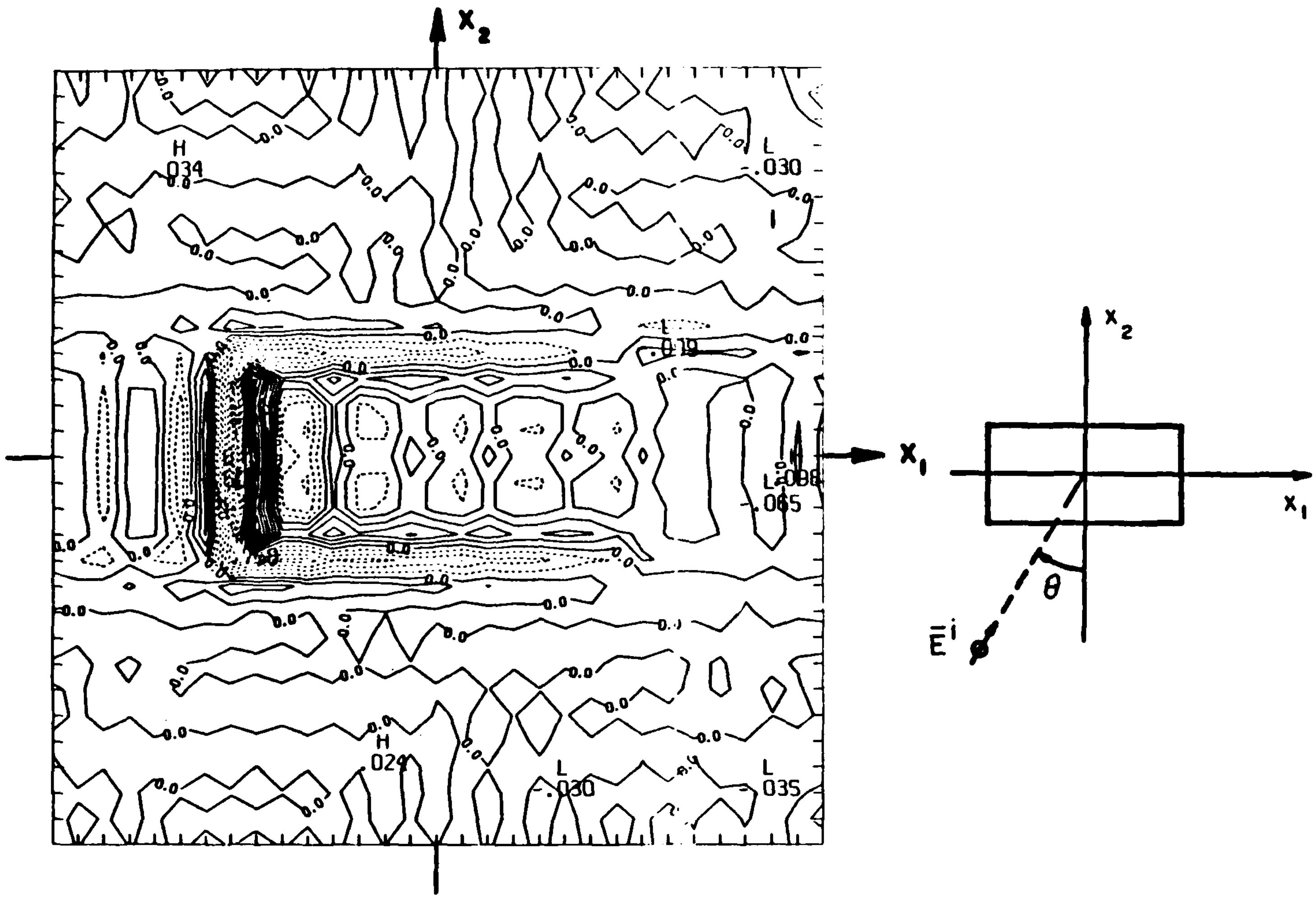


Figure 5-3a. Contour plot of Figure 5-3

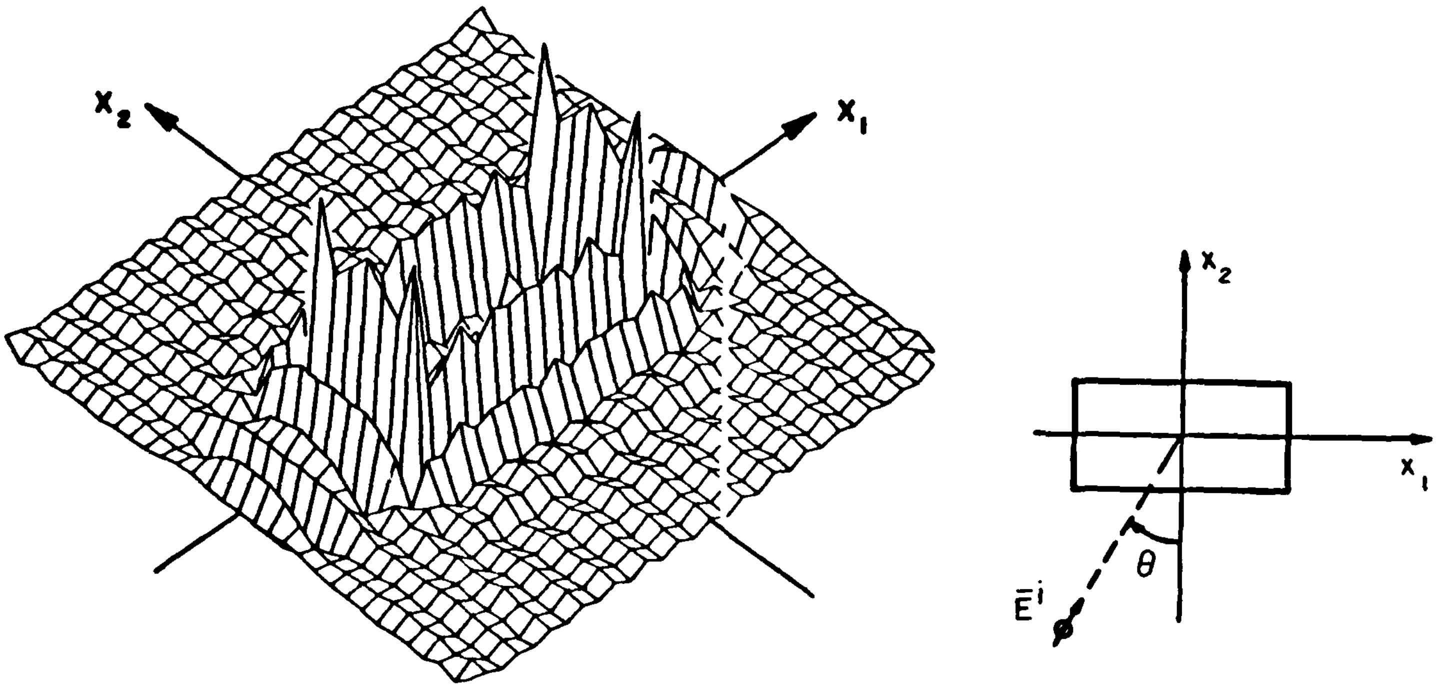


Figure 5-4. Similar description as Figure 5-3, except the data are taken over the full 360° plane

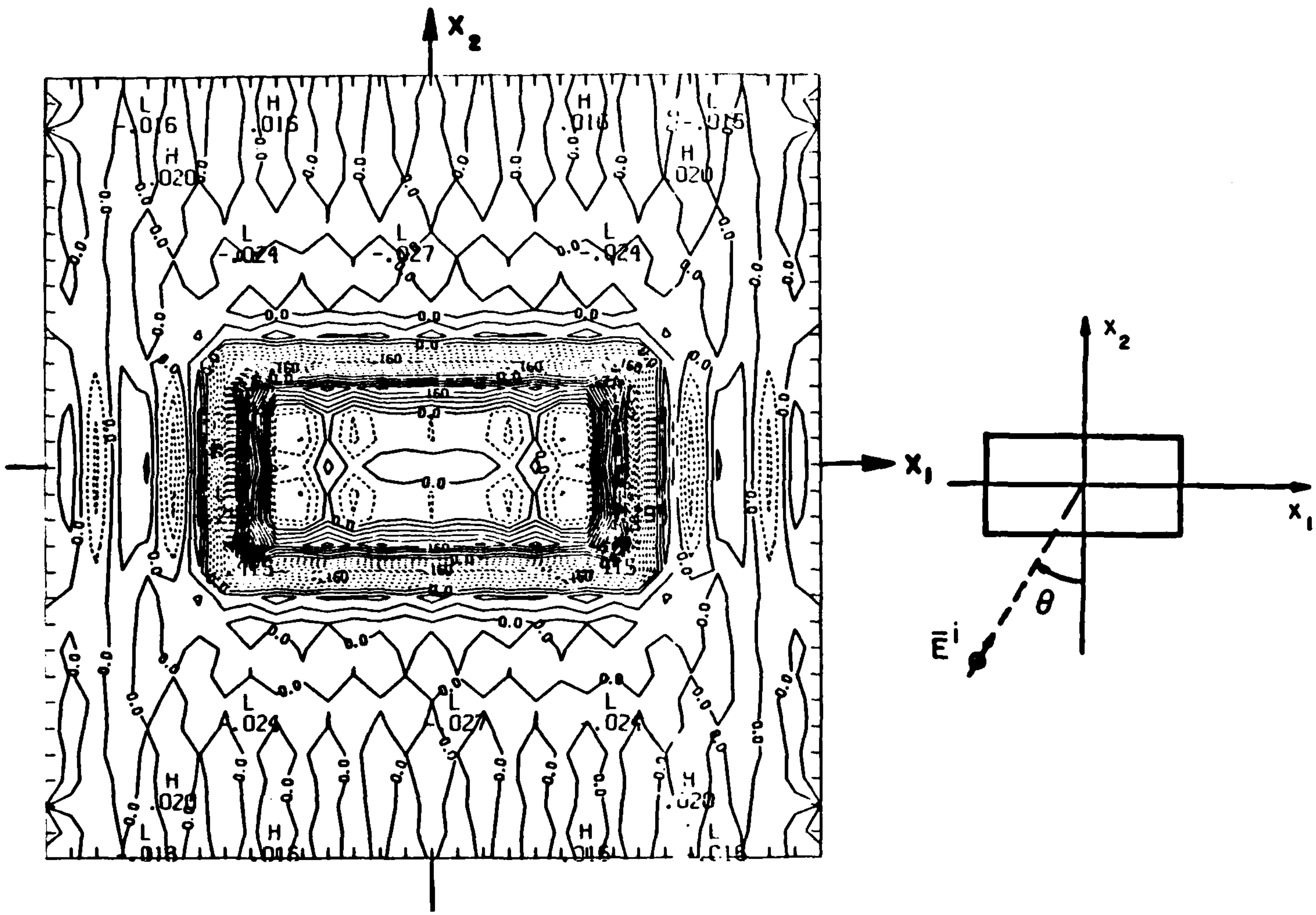


Figure 5-4a. Contour plot of Figure 5-4

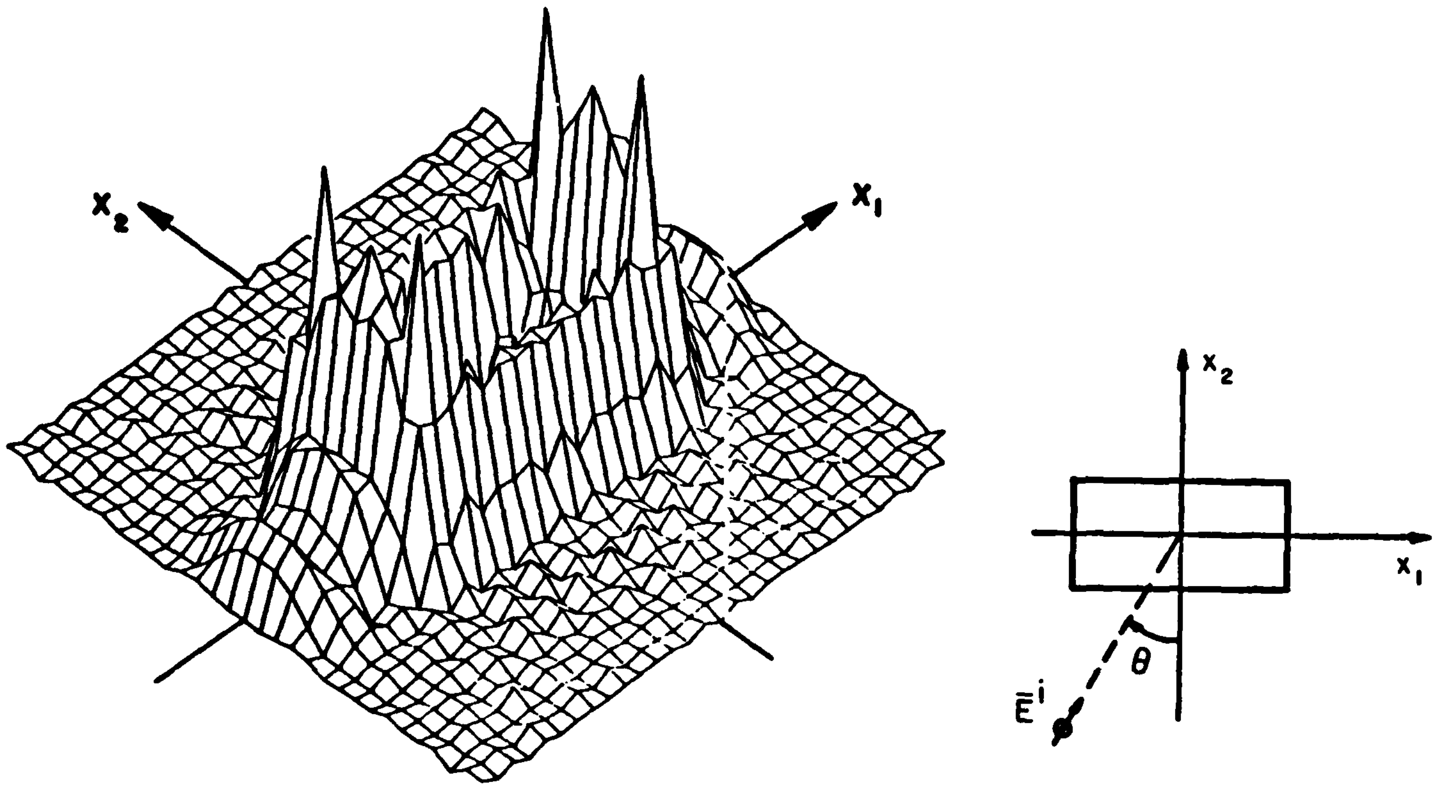


Figure 5-5. The plot of the absolute value of the amplitude in Figure 5-4

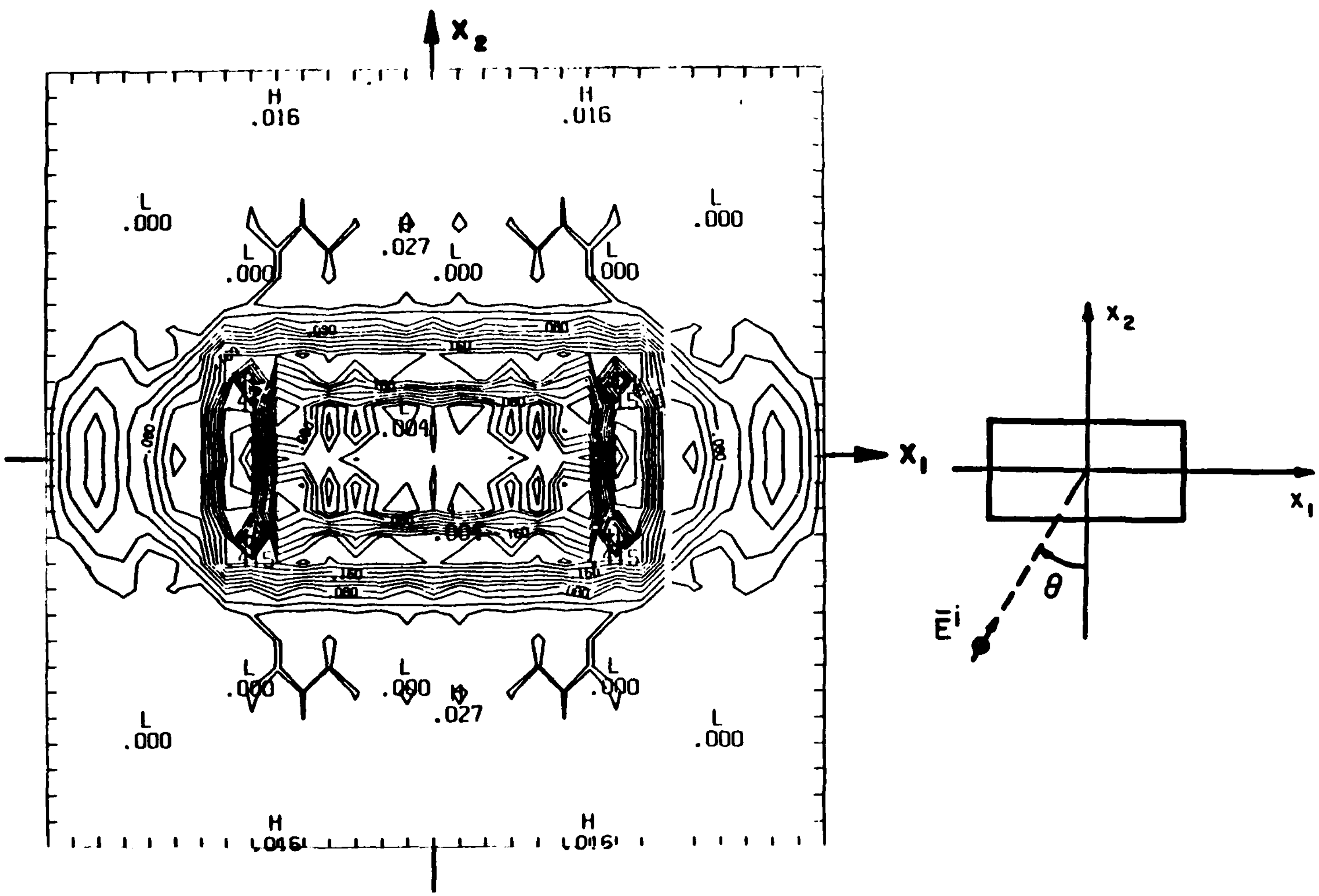


Figure 5-5a. Contour plot of Figure 5-5

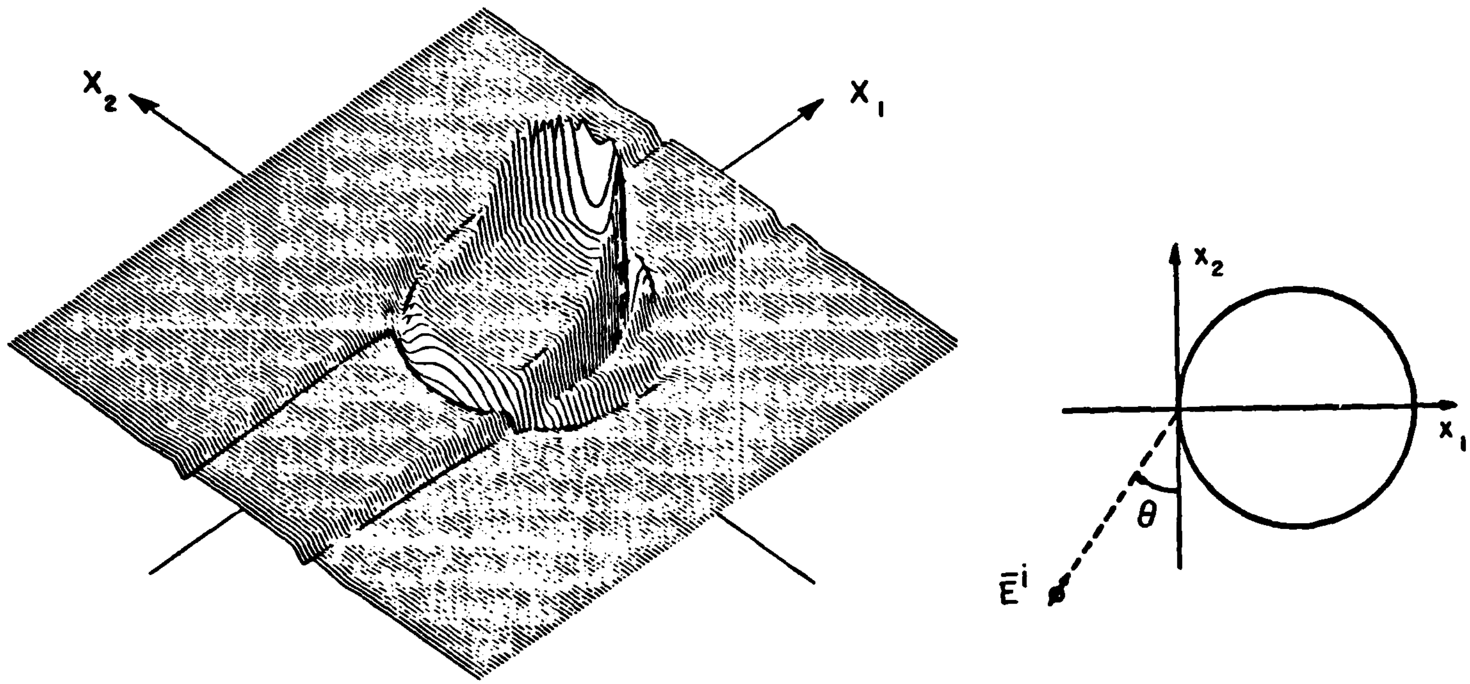


Figure 5-6. 2-D impulse response of a 6" metallic sphere with its centre situated at (3", 0). Frequency samples are taken at the cubic sampling lattice over the range of 0 to 12 Ghz. The data are cosine tapering weighted before Fourier transformed discretely into the spatial doamin. Data are taken over the half plane: $\pi < \theta < 2\pi$

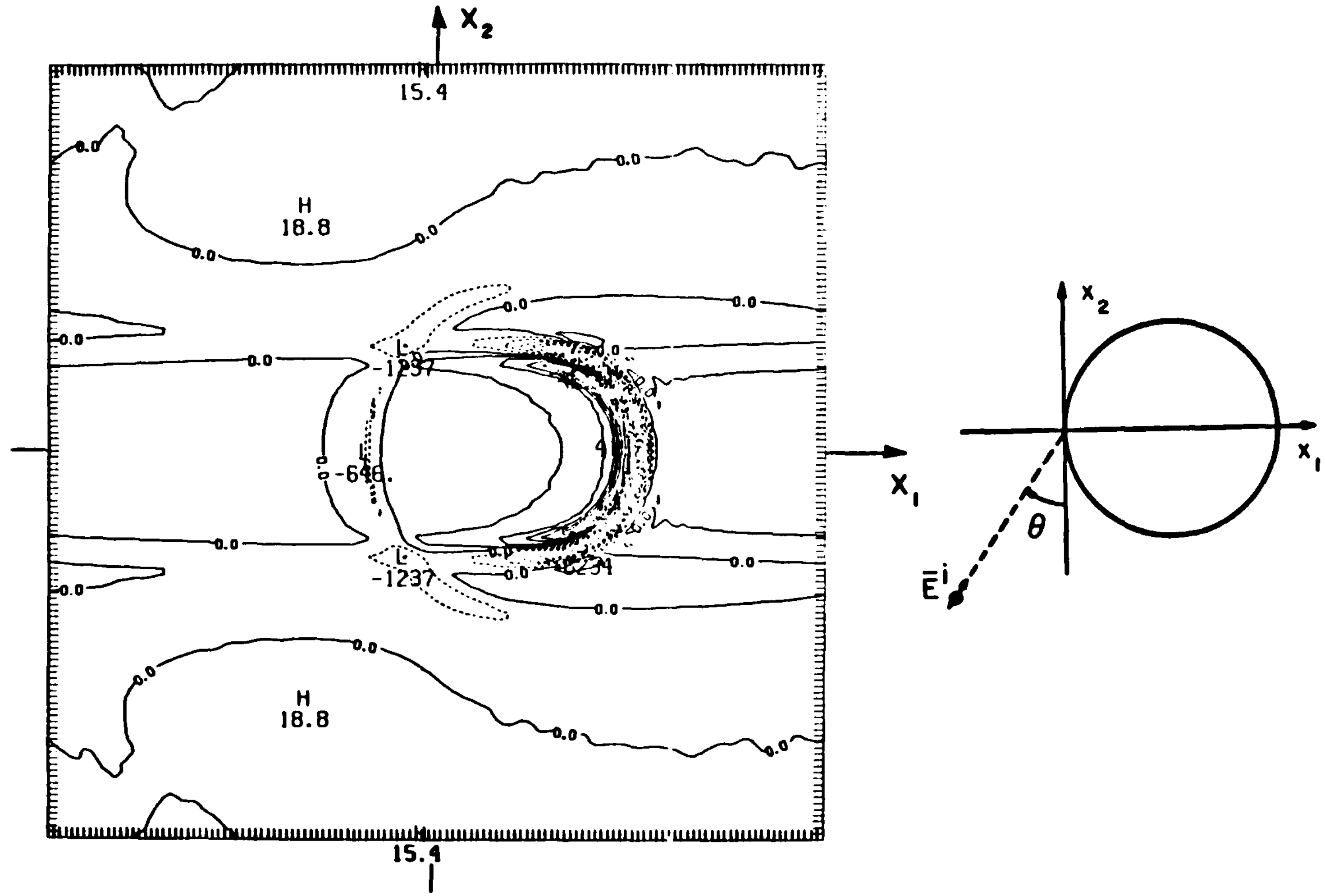


Figure 5-6a. Contour plot of Figure 5-6

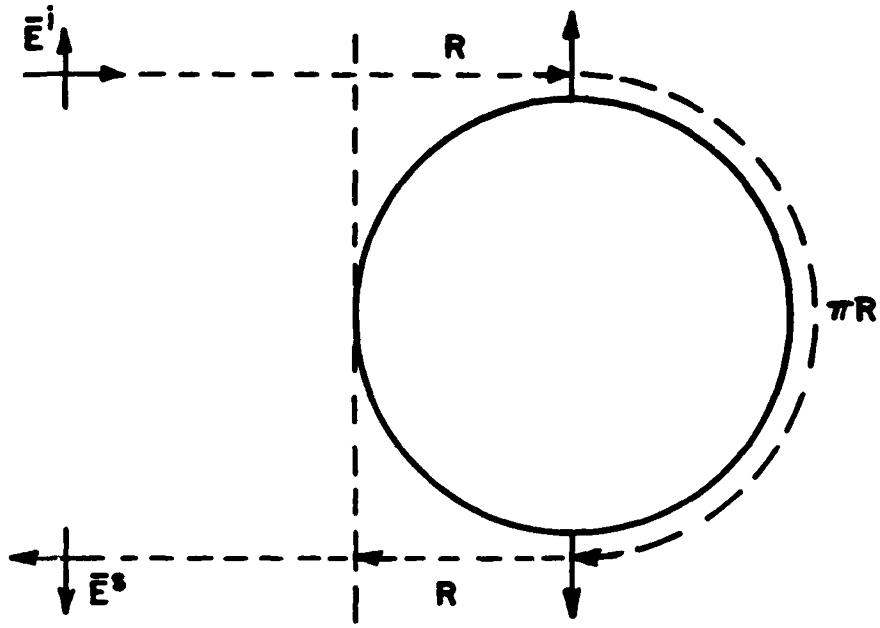


Figure 5-7. Path length of a creeping wave on a metallic sphere

CHAPTER VI

CONCLUSIONS

If a signal is limited in the spatial domain (or wave number domain), this signal is sufficiently characterized by its values over the discrete sampling lattice in the wave number domain (or the spatial domain). In the two dimensional case, the sampling locations in the wave number space are specified by the vector:

$$[l_1\bar{u}_1 + l_2\bar{u}_2]$$

where

$$l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots$$

The $\bar{u}_1$ and $\bar{u}_2$ are related to $\bar{v}_1$ and $\bar{v}_2$ by the following:

$$[\bar{u}_1|\bar{u}_2] = 2\pi[\bar{v}_1|\bar{v}_2]^{-T}$$

where

-T :the transpose of the inverse of the matrix formed

The $\bar{v}_1$ and $\bar{v}_2$ are specified by one's choice on how the space limited signal is assumed to repeat itself in its domain. $\bar{v}_1$ is pointed to the centre of one of the closest images in the periodic lattice. $\bar{v}_2$ is independent of $\bar{v}_1$, and it points at the centre of another close image. By defining the settling time as the end of the impulse response, one has a space limited three dimensional impulse response signal. In the finite circular cylinder data presented in Chapter V, the signal is assumed to be 4 times the size of the cylinder in the two dimensional spatial plane. The choice of 4 is equivalent to choosing the one dimensional impulse response of having a settling time four times as long as the wave would travel over the length of the object at any aspect angle. Or the safety factor is chosen to be one. (i.e., $2(1+K) = 4 \Leftrightarrow K = 1$) This factor provides sufficient results in both Chapter IV and V.

In Chapter V, the example of the finite circular cylinder with dimensions: 6" in length and 3" in diameter, the two dimensional containment unit is chosen to be a square. The squared repetitive lattice in the spatial domain is defined by,

$$\bar{v}_1 = R\begin{bmatrix}1\\0\end{bmatrix} \qquad \bar{v}_2 = R\begin{bmatrix}0\\1\end{bmatrix}$$

where

$$R = 6"$$

and the safety factor K is chosen to be 1. With these inputs to the program PTGRID (see Appendix B), the program outputs the sampling lattice defined by $\bar{u}_1$ and $\bar{u}_2$. The data output is arranged with the

aspect angles and the corresponding frequency increment. One makes the necessary data extraction at the proper aspect angles, and increments the frequency until the highest frequency is reached. One could also have defined a rectangular repetitive lattice. Then

$$\bar{v}_1 = R \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \bar{v}_1 = R \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

where

$$R = 6"$$

Or, for a circular repetitive lattice,

$$\bar{v}_1 = R \begin{bmatrix} \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{bmatrix} \qquad \bar{v}_2 = R \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where

$$R = \sqrt{(6")^2 + (3")^2}$$

$$= 6.708"$$

Again these $\bar{v}_1$ and $\bar{v}_2$ values can be input into the program PTGRID (see Appendix B) to obtain the sampling lattice.

Thus, one can sample discretely and interpolate in the wave number space to reproduce the frequency response of an object. Fourier transforming this wave number signal into the spatial domain gives a representation of the spatial impulse response of the object. The

spatial impulse response is defined as the image function obtained by three dimensional Fourier transforming the far field frequency response of a finite object at all aspect angles defined over the 4π solid angle. Since most objects have different shapes, sizes, and orientations, their corresponding spatial impulse responses are different. Their corresponding sampling lattices will also be different. One general sampling lattice that is applicable to a set of objects is desirable. This is accomplished by introducing different types of canonical containment units to confine the spatial impulse responses. Two common canonical cells are parallelepiped and sphere. Two dimensional examples are shown in Figures 2-5 and 2-7, with their corresponding sampling lattices shown in Figures 2-6 and 2-8 respectively.

In Chapter IV, a six inch metallic sphere is chosen as an example to compare two types of sampling lattices -cubic and isotropic. The comparison is performed on the interpolated one dimensional impulse responses at different aspect angles using the interpolation scheme defined in the sampling theorem. As one expects, the results turn out to be competitive for the two types of sampling. Efficiencies, in the sense of the least number of sampling points, are different for different canonical containment cells used on the same object.

Efficiency in using one type of canonical containment cell:

$$\eta = \frac{\text{CONSERVATIVE ESTIMATE OF THE OBJECT'S VOLUME}}{\text{VOLUME OF THE SMALLEST CANONICAL CONTAINMENT CELL ENCLOSING THE OBJECT}}$$

Efficiencies among a sphere cap cylinder, a sphere and a cube using cubic, isotropic, and rectangular box confinement units are presented in Chapter IV. The efficiency definition proves to be a very good concept in deciding the type of sampling lattice for an object or a group of objects. This is under the assumption that the settling time of the spatial impulse response is shaped similarly to the object; e.g., the settling time of the impulse response of a sphere is the same in every aspect angle.

To obtain an approximation to the spatial impulse response from the sampled data over a finite frequency range, one can use the discrete Fourier transform. Because of today's digital computer design, the different sufficient characterization cannot be readily processed without interpolation; except, of course, the cubic lattice data sets. The interpolation step which most people like to avoid, is very time consuming. This may be referred back to the time estimation example on a sphere cap cylinder presented in Chapter V. The interpolation step is another factor that affects an engineering decision. The avoidance helps Mensa et al. to arrive at the time response faster. The price they paid is the limitation of their method's application to two dimensional Fourier transform. Their approach is thus not recommended because of its inability to be expanded into higher dimensions. The sampling criteria accompanying their method are,

the angular increment:

$$\Delta\theta \begin{cases} < \dfrac{\lambda}{2D} & \lambda << 2D \\ < \sin^{-1}(\dfrac{\lambda}{2D}) & \lambda < 2D \\ < \dfrac{\pi}{4} & \lambda > 2D \end{cases}$$

the frequency increment:

$$\Delta f < \frac{c}{2(1+K)D}$$

where

D = maximum dimension of the object

λ = wavelength of the frequency used

c = speed of light

K = some safety factor

In the finite circular cylinder example,

$D = 17.04$ cm

$K = 0.75$

$\Delta f < 0.503$ Ghz

at the highest frequency of 10 Ghz,

$\lambda = 3$ cm

$\Delta\theta < 5$ degrees

Therefore,

the frequency increment chosen = 0.5 Ghz

the angular increment chosen = 1 degree

Nevertheless, processed results from UTD solution on a finite circular cylinder support the convergence of Mensa et al.'s method to two dimensional discrete Fourier transform.

Lewis-Bojarski's identity requires either the object be illuminated at all angles or assumption be made on the shadowed side. Results on a finite metallic circular cylinder and a metallic sphere indicate that the spatial impulse response approach does not have the above restriction, though proper interpretation may be required. Furthermore, the use of the spatial impulse response to imaging can converge to Lewis-Bojarski's results. There is also an indication that the presentation in the form of the absolute value of the amplitude does not necessarily provide the proper picture for pattern recognition. The plain amplitude representation with positive and negative values is sometimes more appropriate. This is concluded by comparing the Figure 5-4, or 5-4a with 5-5 or 5-5a. The perimeter of a major axis cross-section of a finite circular cylinder is shown more distinctly using the plain amplitude presentation. Judging from the two dimensional impulse responses, one can deduce the substantial potential of the spatial impulse response in image reconstruction.

The spatial impulse response has numerous applications including target identification and imaging. Although smooth convex metallic body examples are considered here, tomographic applications on other types of bodies are possible. In all, the N dimensional sampling theorem provides new insights into the sampling criteria in the wave number space for a finite object. The potential in reduced management time on

two or three dimensional data is enormous. The two dimensional impulse response also projects a promising target imaging technique using the spatial impulse response.

CHAPTER VII

RECOMMENDATIONS

Traditionally, two and three dimensional data are presented in cubic lattices, for which present day digital computers are designed. Although the digital computer of today manages data in cubic lattices, the cubic lattices are not necessarily the most efficient in terms of the least number of data samples. The most general approach to solve this computer problem requires the cubic data management structure of the digital computer be modified into a more general data structure. In another words, data organized in any random fashion can be processed by this computer. If this general approach is not practical, the next best step is a faster interpolation scheme either in hardware or software. The lowest level on the hierarchy of improvement is the improvement for specific application. In this thesis, a general Fourier transform that can perform on any data lattice, fits into this category.

After some of the computer problems are solved, the next step in the development is to account for the experimental noise. How does one extract the true information that is embedded in noise? Without this

generalization, this report is only useful in information storage and retrival on noise free data. Noise free is in the sense that the noise effect in measurement is eliminated before storage. The full development of this report's theory will enable significant reduction in time on data measurement, processing and storage.

The whole report is focused on the monostatic data. It would be interesting to see how this theory holds up with the bistatic data. Using the definition of the three dimensional Fourier transform, one can derive a similar method that parallels Mensa et al.'s approach.

By converting the k-space coordinates into spherical coordinates:

$$k_1 = \rho\sin\theta\cos\phi \qquad k_2 = \rho\sin\theta\sin\phi \qquad k_3 = \rho\cos\theta \tag{7-1}$$

equation (1-1) becomes:

$$f(\bar{x}) = \frac{1}{8\pi^3} \int_{\rho=0}^{\infty} \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \tilde{F}(\rho, \theta, \phi) e^{j|\bar{x}|\rho\cos\angle(\bar{x}, \rho)} \rho^2 \sin\theta \, d\phi \, d\theta \, d\rho \tag{7-2}$$

where

$$|\bar{x}| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

(Rotate the k-space coordinate system so that the k_3-axis is in line with $\bar{x}$: => $\angle(\bar{x}, \rho) = \theta$).

Then,

$$f(\bar{x}) = \frac{1}{8\pi^3} \int_{\rho=0}^{\infty} \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \tilde{F}(\rho, \theta, \phi) e^{j|\bar{x}|\rho\cos\theta} \rho^2 \sin\theta d\phi d\theta d\rho \tag{7-3}$$

By considering one particular aspect angle: ϕ_m, θ_i

$$f_{im}(\bar{x}) = \frac{\sin\theta_i}{8\pi^3} \int_{\rho=0}^{\infty} \tilde{F}(\rho, \theta_i, \phi_m) e^{j|\bar{x}|\rho\cos\theta_i} \rho^2 d\rho \tag{7-4}$$

With all aspect angular data,

$$f(\bar{x}) = \sum_{i=1}^{\infty} \sum_{m=1}^{\infty} f_{im}(\bar{x}) \tag{7-5}$$

$$= \text{the impulse response of the finite object at } \bar{x}$$

Even though Equation (7-4) requires the integration over all frequencies, the integral can be approximated over three regions: the Rayleigh, the resonance, and the optical. Thus, the integral in Equation (7-4) is not impossible to be solved. There are problems associated with this approach. The Nyquist angular requirement is frequency dependent (Equation (2-17)). In order to satisfy the Nyquist angular requirement at high frequency, the signal is excessively sampled at the low frequency spectrum, but sampled only adequately at the high frequency spectrum for a finite frequency range of interest. Nonetheless, the approach is viable if one does not intend to extend the bandwidth of the approximated spatial impulse response.

With the help of the different sampling criteria, the infinite number of samples required to reconstruct the impulse response over a finite frequency range has changed to a finite number. Though the number is finite, the measurement time can be very substantial when an expanded frequency range in two dimensions, or three dimensions is required. Professor Leon Peters suggested another approach to further reduce the measurement time [15]. At high frequency, a target's frequency response is mostly contributed by its major scattering centres. If one can make a set of different canonical scattering centre measurement, then most targets' high frequency response can be built using the proper phase shift factors. For most scattering centres, their frequency responses are relatively simple. As a result, the Nyquist criteria for these centres in the frequency domain are more relaxed than complete structures. These canonical scattering centre data can be reused to reconstruct the high frequency response of other targets. In another words, after the canonical scattering centre data are available, one only measures the low frequency spectrum before the one dimensional, two dimensional, or three dimensional impulse response of an object can be reconstructed. This would be another interesting area for further exploration. However, more development work is required in all these described areas to extend this report into a more useful engineering tool.

APPENDIX A

THE DERIVATION OF EQUATION (2-13)

$$G(\omega_1, \omega_2) = \frac{1}{R^2\omega_1(\omega_1^2 - 3\omega_2^2)} \times \left\{ \begin{array}{l} 2\omega_1 \cos\left(\frac{R\omega_1}{\sqrt{3}}\right) \cos(R\omega_2) \\ -2\omega_1 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \\ -2\sqrt{3}\, \omega_2 \sin\left(\frac{R\omega_1}{\sqrt{3}}\right) \sin(R\omega_2) \end{array} \right\}$$

Equation (65) of Petersen and Middleton [7]:

$$G(\bar{\omega}) = \left(\frac{1}{R^2}\right)\left(\frac{1}{2\sqrt{3}}\right) \iint_{\substack{\text{Regular} \\ \text{Hexagon}}} e^{j\bar{\omega}\cdot\bar{x}}\, dx$$

Equation description for the regular hexagon: (see Figure A-1)

$$① \left\{ \begin{array}{l} x_2 = R \\ \left(\frac{-R}{\sqrt{3}}\right) < x_1 < \left(\frac{R}{\sqrt{3}}\right) \end{array} \right.$$

$$② \left\{ \begin{array}{l} x_2 = -\sqrt{3}\, x_1 + 2R \\ \left(\frac{R}{\sqrt{3}}\right) < x_1 < \left(\frac{2R}{\sqrt{3}}\right) \end{array} \right.$$

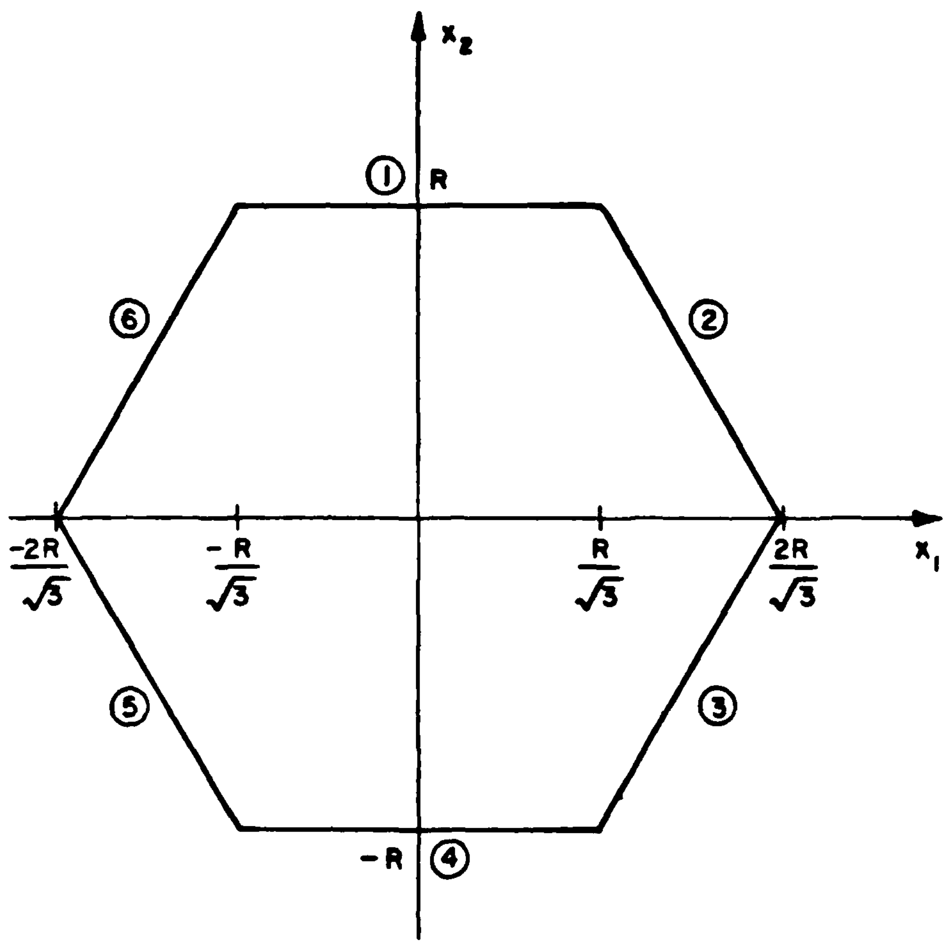


Figure A-1. Hexagonal integration limit

$$③\begin{cases} x_2 = \sqrt{3}\,x_1 - 2R \\ \left(\frac{R}{\sqrt{3}}\right) < x_1 < \left(\frac{2R}{\sqrt{3}}\right) \end{cases}$$

$$④\begin{cases} x_2 = -R \\ \left(\frac{-R}{\sqrt{3}}\right) < x_1 < \left(\frac{R}{\sqrt{3}}\right) \end{cases}$$

$$⑤\begin{cases} x_2 = -\sqrt{3}\,x_1 - 2R \\ \left(\frac{-2R}{\sqrt{3}}\right) < x_1 < \left(\frac{-R}{\sqrt{3}}\right) \end{cases}$$

$$⑥\begin{cases} x_2 = \sqrt{3}\,x_1 + 2R \\ \left(\frac{-2R}{\sqrt{3}}\right) < x_1 < \left(\frac{-R}{\sqrt{3}}\right) \end{cases}$$

then

$$G(\bar{\omega}) = \frac{1}{2\sqrt{3}R^2}\,(I_1 + I_2 + I_3) \qquad \text{(A-1)}$$

where

$$I_1 = \int_{x_1=\left(\frac{-2R}{\sqrt{3}}\right)}^{\left(\frac{-R}{\sqrt{3}}\right)} \int_{x_2=⑤}^{⑥} e^{j(\omega_1 x_1 + \omega_2 x_2)}\,dx_2\,dx_1 \qquad \text{(A-2)}$$

$$I_2 = \int_{x_1=\left(\frac{-R}{\sqrt{3}}\right)}^{\left(\frac{R}{\sqrt{3}}\right)} \int_{x_2=-R}^{R} e^{j(\omega_1 x_1 + \omega_2 x_2)}\,dx_2\,dx_1 \qquad \text{(A-3)}$$

$$I_3 = \int_{x_1 = \left(\frac{R}{\sqrt{3}}\right)}^{\left(\frac{2R}{\sqrt{3}}\right)} \int_{x_2 = 3}^{2} e^{j(\omega_1 x_1 + \omega_2 x_2)} dx_2\, dx_1 \qquad \text{(A-4)}$$

Equation (A-2):

$$I_1 = \int_{x_1 = \left(\frac{-2R}{\sqrt{3}}\right)}^{\left(\frac{-R}{\sqrt{3}}\right)} \int_{x_2 = -\sqrt{3}x_1 - 2R}^{\sqrt{3}x_1 + 2R} e^{j\omega_1 x_1} \cdot e^{j\omega_2 x_2} dx_2\, dx_1$$

$$= \int_{x_1 = \left(\frac{-2R}{\sqrt{3}}\right)}^{\left(\frac{-R}{\sqrt{3}}\right)} e^{j\omega_1 x_1} \left[\frac{e^{j\omega_2 x_2}}{j\omega_2}\right]_{-(\sqrt{3}x_1 + 2R)}^{(\sqrt{3}x_1 + 2R)} dx_1$$

$$= \frac{1}{j\omega_2} \int_{x_1 = \left(\frac{-2R}{\sqrt{3}}\right)}^{\left(\frac{-R}{\sqrt{3}}\right)} e^{j\omega_1 x_1} \left\{ e^{j\omega_2(\sqrt{3}x_1 + 2R)} - e^{-j\omega_2(\sqrt{3}x_1 + 2R)} \right\} dx_1$$

$$= \frac{1}{j\omega_2} \int_{x_1 = \left(\frac{-2R}{\sqrt{3}}\right)}^{\left(\frac{-R}{\sqrt{3}}\right)} \left\{ e^{j(\omega_1 + \sqrt{3}\omega_2)x_1 + 2jR\omega_2} - e^{j(\omega_1 - \sqrt{3}\omega_2)x_1 - 2j\omega_2 R} \right\} dx_1$$

$$= \frac{1}{j\omega_2} \left\{ \left[\frac{e^{j(\omega_1 + \sqrt{3}\omega_2)x_1 + 2jR\omega_2}}{j(\omega_1 + \sqrt{3}\omega_2)} \right]_{\frac{-2R}{\sqrt{3}}}^{\frac{-R}{\sqrt{3}}} \right.$$

$$\left. - \left[\frac{e^{j(\omega_1 - \sqrt{3}\omega_2)x_1 - 2jR\omega_2}}{j(\omega_1 - \sqrt{3}\omega_2)} \right]_{\frac{-2R}{\sqrt{3}}}^{\frac{-R}{\sqrt{3}}} \right\}$$

Therefore,

$$I_1 = \frac{e^{2jR\omega_2}}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left[e^{-j(\omega_1 + \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}})} - e^{-j(\omega_1 + \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})} \right]$$

$$+ \frac{e^{-2jR\omega_2}}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left[e^{-j(\omega_1 - \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})} - e^{-j(\omega_1 - \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}})} \right]$$

Equation (A-3):

$$I_2 = \int_{x_1 = \frac{-R}{\sqrt{3}}}^{\frac{R}{\sqrt{3}}} \int_{x2 = -R}^{R} e^{j(\omega_1 x_1 + \omega_2 x_2)} dx_2 \, dx_1$$

$$= \int_{x_1 = \left(\frac{-R}{\sqrt{3}}\right)}^{\left(\frac{R}{\sqrt{3}}\right)} e^{j\omega_1 x_1} dx_1 \int_{x2 = -R}^{R} e^{j\omega_2 x_2} dx_2$$

$$= \left[\frac{e^{j\omega_1 x_1}}{j\omega_1}\right]_{\frac{-R}{\sqrt{3}}}^{\frac{R}{\sqrt{3}}} \left[\frac{e^{j\omega_2 x_2}}{j\omega_2}\right]_{-R}^{R}$$

$$= \left[\left(\frac{2}{\omega_1}\right) \sin\left(\frac{\omega_1 R}{\sqrt{3}}\right)\right]\left[\left(\frac{2}{\omega_2}\right) \sin(\omega_2 R)\right]$$

Therefore,

$$I_2 = \left(\frac{4}{\omega_1 \omega_2}\right) \sin(\omega_2 R) \sin\left(\frac{\omega_1 R}{\sqrt{3}}\right)$$

Equation (A-4),

$$I_3 = \int_{x_1=\left(\frac{R}{\sqrt{3}}\right)}^{\left(\frac{2R}{\sqrt{3}}\right)} \int_{x_2=\sqrt{3}x_1 - 2R}^{-\sqrt{3}x_1 + 2R} e^{j(\omega_1 x_1 + \omega_2 x_2)} dx_2 dx_1$$

$$= \int_{x_1=\left(\frac{R}{\sqrt{3}}\right)}^{\left(\frac{2R}{\sqrt{3}}\right)} e^{j\omega_1 x_1} \left[\frac{e^{j\omega_2 x_2}}{j\omega_2}\right]_{\sqrt{3}x_1 - 2R}^{-\sqrt{3}x_1 + 2R} dx_1$$

$$= \frac{1}{j\omega_2} \int_{x_1=\left(\frac{R}{\sqrt{3}}\right)}^{\left(\frac{2R}{\sqrt{3}}\right)} e^{j\omega_1 x_1} \left[e^{j\omega_2(-\sqrt{3}x_1 + 2R)} - e^{j\omega_2(\sqrt{3}x_1 - 2R)}\right] dx_1$$

$$= \frac{1}{j\omega_2} \int_{x_1=\left(\frac{R}{\sqrt{3}}\right)}^{\left(\frac{2R}{\sqrt{3}}\right)} \left[e^{j(\omega_1 - \sqrt{3}\omega_2)x_1 + 2jR\omega_2} - e^{j(\omega_1 + \sqrt{3}\omega_2)x_1 - 2jR\omega_2}\right] dx_1$$

$$= \frac{1}{j\omega_2} \left\{ \left[\frac{e^{j(\omega_1 - \sqrt{3}\omega_2)x_1 + 2jR\omega_2}}{j(\omega_1 - \sqrt{3}\omega_2)} \right]_{\frac{R}{\sqrt{3}}}^{\frac{2R}{\sqrt{3}}} \right.$$

$$\left. - \left[\frac{e^{j(\omega_1 + \sqrt{3}\omega_2)x_1 - 2jR\omega_2}}{j(\omega_1 + \sqrt{3}\omega_2)} \right]_{\frac{R}{\sqrt{3}}}^{\frac{2R}{\sqrt{3}}} \right\}$$

Therefore,

$$I_3 = \frac{e^{2jR\omega_2}}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left[e^{j(\omega_1 - \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})} - e^{j(\omega_1 - \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}})} \right]$$

$$+ \frac{e^{-2jR\omega_2}}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left[e^{j(\omega_1 + \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}})} - e^{j(\omega_1 + \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})} \right]$$

Now,

$$I_1 + I_3$$

$$= \frac{1}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ e^{-j[(\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2]} + e^{j[(\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2]} \right.$$

$$\left. - e^{-j[(\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2]} - e^{j[(\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2]} \right\}$$

$$+ \frac{1}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ e^{-j[(\omega_1 - \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} + 2R\omega_2]} + e^{j[(\omega_1 - \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} + 2R\omega_2]} \right.$$

$$\left. - e^{-j[(\omega_1 - \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} + 2R\omega_2]} - e^{j[(\omega_1 - \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} + 2R\omega_2]} \right\}$$

$$= \frac{1}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ 2\cos\left[(\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2\right] \right.$$

$$\left. - 2\cos\left[(\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2\right] \right\}$$

(This expression continues on the next page.)

$$+ \frac{1}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ 2\cos\left[(\omega_1 - \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} + 2R\omega_2\right]\right.$$

$$\left. - 2\cos\left[(\omega_1 - \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} + 2R\omega_2\right]\right\}$$

$$\left[\text{Since:}\quad \cos\alpha - \cos\beta = -2\sin\frac{1}{2}(\alpha+\beta)\sin\frac{1}{2}(\alpha-\beta)\right]$$

$$= \frac{-4}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ \sin\frac{1}{2}\left[\left(\frac{3R}{\sqrt{3}}\right)(\omega_1 + \sqrt{3}\omega_2) - 4R\omega_2\right] \sin\frac{1}{2}\left[\left(\frac{R}{\sqrt{3}}\right)(\omega_1 + \sqrt{3}\omega_2)\right]\right\}$$

$$+ \frac{4}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ \sin\frac{1}{2}\left[\left(\frac{3R}{\sqrt{3}}\right)(\omega_1 - \sqrt{3}\omega_2) + 4R\omega_2\right] \sin\frac{1}{2}\left[\left(\frac{R}{\sqrt{3}}\right)(\omega_1 - \sqrt{3}\omega_2)\right]\right\}$$

$$= \frac{-4}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ \sin\left(\frac{\sqrt{3}R\omega_1}{2} - \frac{R\omega_2}{2}\right) \sin\left(\frac{R\omega_1}{2\sqrt{3}} + \frac{R\omega_2}{2}\right)\right\}$$

$$+ \frac{4}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ \sin\left(\frac{R\sqrt{3}\omega_1}{2} + \frac{R\omega_2}{2}\right) \sin\left(\frac{R\omega_1}{2\sqrt{3}} - \frac{R\omega_2}{2}\right)\right\}$$

$$\left[\text{Since: } \sin\alpha\sin\beta = \frac{1}{2}\cos(\alpha-\beta) - \frac{1}{2}\cos(\alpha+\beta)\right]$$

$$= \frac{-2}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{\cos\left[\left(\sqrt{3} - \frac{1}{\sqrt{3}}\right)\left(\frac{R\omega_1}{2}\right) - R\omega_2\right] - \cos\left[\left(\sqrt{3} + \frac{1}{\sqrt{3}}\right)\frac{R\omega_1}{2}\right]\right\}$$

$$+ \frac{2}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{\cos\left[\left(\sqrt{3} - \frac{1}{\sqrt{3}}\right)\left(\frac{R\omega_1}{2}\right) + R\omega_2\right] - \cos\left[\left(\sqrt{3} + \frac{1}{\sqrt{3}}\right)\frac{R\omega_1}{2}\right]\right\}$$

$$= \frac{-2}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{\cos\left(\frac{R\omega_1}{\sqrt{3}} - R\omega_2\right) - \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right)\right\}$$

$$+ \frac{2}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{\cos\left(\frac{R\omega_1}{\sqrt{3}} + R\omega_2\right) - \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right)\right\}$$

$$= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ (\omega_1 - \sqrt{3}\omega_2) \cos\left(\frac{R\omega_1}{\sqrt{3}} - R\omega_2\right) \right.$$

$$- (\omega_1 - \sqrt{3}\omega_2) \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right)$$

$$- (\omega_1 + \sqrt{3}\omega_2) \cos\left(\frac{R\omega_1}{\sqrt{3}} + R\omega_2\right)$$

$$\left. + (\omega_1 + \sqrt{3}\omega_2) \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \right\}$$

$$= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \omega_1 \left[\cos\left(\frac{R\omega_1}{\sqrt{3}} - R\omega_2\right) - \cos\left(\frac{R\omega_1}{\sqrt{3}} + R\omega_2\right)\right] \right.$$

$$- \sqrt{3}\,\omega_2 \left[\cos\left(\frac{R\omega_1}{\sqrt{3}} - R\omega_2\right) + \cos\left(\frac{R\omega_1}{\sqrt{3}} + R\omega_2\right)\right]$$

$$\left. + 2\sqrt{3}\,\omega_2 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \right\}$$

$$\left[\begin{array}{ll} \text{Since:} & \cos\alpha + \cos\beta = 2\cos\frac{1}{2}(\alpha+\beta)\cos\frac{1}{2}(\alpha-\beta) \\ & \cos\alpha - \cos\beta = -2\sin\frac{1}{2}(\alpha+\beta)\sin\frac{1}{2}(\alpha-\beta) \end{array}\right]$$

$$= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \omega_1 \left[2 \sin \frac{1}{2}\left(\frac{2R\omega_1}{\sqrt{3}}\right) \sin \frac{1}{2} (2R\omega_2) \right] \right.$$

$$- \sqrt{3}\ \omega_2 \left[2 \cos \frac{1}{2}\left(\frac{2R\omega_1}{\sqrt{3}}\right) \cos \frac{1}{2} (-2R\omega_2) \right]$$

$$\left. + 2\sqrt{3}\, \omega_2 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \right\}$$

Therefore, $I_1 + I_3$

$$= \frac{-4}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \omega_1 \sin\left(\frac{R\omega_1}{\sqrt{3}}\right) \sin (R\omega_2) \right.$$

$$- \sqrt{3}\, \omega_2 \cos\left(\frac{R\omega_1}{\sqrt{3}}\right) \cos (R\omega_2)$$

$$\left. + \sqrt{3}\, \omega_2 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \right\}$$

Now, $I_1 + I_2 + I_3$

$$= \frac{-4}{\omega_1\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \sqrt{3}\omega_2\omega_1 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) + \omega_1^2 \sin\left(\frac{R\omega_1}{\sqrt{3}}\right) \sin (R\omega_2) \right.$$

$$- \sqrt{3}\, \omega_1\omega_2 \cos\left(\frac{R\omega_1}{\sqrt{3}}\right) \cos (R\omega_2)$$

$$- \omega_1^2 \sin (\omega_2 R) \sin\left(\frac{\omega_1 R}{\sqrt{3}}\right)$$

$$\left. + 3\, \omega_2^2 \sin (\omega_2 R) \sin\left(\frac{\omega_1 R}{\sqrt{3}}\right) \right\}$$

$$= \frac{-4}{\omega_1 \left(\omega_1^2 - 3\omega_2^2\right)} \left\{ \begin{array}{l} \sqrt{3}\,\omega_1 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) - \sqrt{3}\,\omega_1 \cos\left(\frac{R\omega_1}{\sqrt{3}}\right) \cos\left(R\omega_2\right) \\ + 3\,\omega_2 \sin\left(\omega_2 R\right) \sin\left(\frac{\omega_1 R}{\sqrt{3}}\right) \end{array} \right\}$$

Equation (A-1),

$$G(\bar{\omega}) = \frac{1}{2\sqrt{3}R^2} \left(I_1 + I_2 + I_3\right)$$

$$= \frac{1}{R^2 \omega_1^2 \left(\omega_1^2 - 3\omega_2^2\right)} \times \left\{ \begin{array}{l} 2\,\omega_1 \cos\left(\frac{R\omega_1}{\sqrt{3}}\right) \cos\left(R\omega_2\right) \\ - 2\,\omega_1 \cos\left(\frac{2R\omega_1}{\sqrt{3}}\right) \\ - 2\sqrt{3}\,\omega_2 \sin\left(\frac{R\omega_1}{\sqrt{3}}\right) \sin\left(R\omega_2\right) \end{array} \right\}$$

= Equation (2-13).

APPENDIX B

PROGRAMS DEVELOPED

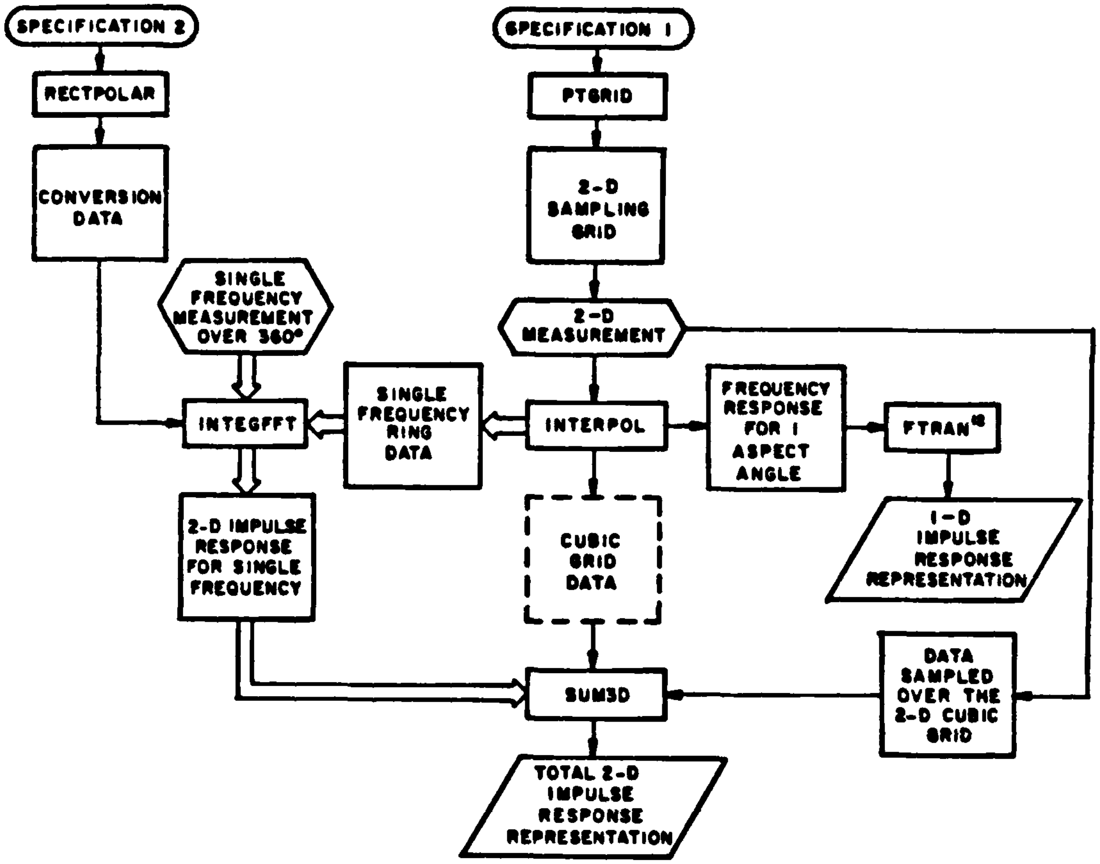


Figure B-1. Flow chart showing the relationship among the written programs

```
C
C     PROGRAM:        RECONST1D
C
C     THIS PROGRAM WILL TAKE A DATA SET:'COEFF' IN REAL AND IMAGINARY FORMAT
C     AND INTERPOLATE THE POINTS IN BETWEEN SAMPLES USING SAMPLING THEOREM
C     APPROACH.  PRESENTLY, THIS
C     PROGRAM IS ALLOWED TO TAKE 100 SAMPLES. IF MORE IS REQUIRED, PLEASE
C     CHANGE THE ARRAY SIZE OF COEFF AND THE VALUE OF NUM.
C             MSIZE:THE ARRAY SIZE=2*(NUMBER OF SAMPLES) +1
C             NUM:NUMBER OF SAMPLES
C             XINC: INCREMENT
C     THE SAMPLES ARE TAKEN AT DELTA FREQUENCY OF 'FREO'.
C     THE FREQUENCY RANGE FOR INTERPOLATION IS SPECIFIED BY 'INITIAL'
C     AND 'LAST'.  THE NUMBER OF POINTS IN 'DATA' IS 'NP1'. THE
C     OUTPUT FILE IS SPECIFIED BY 'FNAME2'.
C
C     THIS PROGRAM WILL LINK WITH SINC, JPLOT, SUM1D AND 'PLOTLIB
C
      REAL INITIAL, LAST
      COMPLEX COEFF(201),SUM,DATA(200)
      CHARACTER*10 FNAME1.FNAME2
      CHARACTER COM1
      NUM=100
      MSIZE=2*NUM +1
      NSTEP=200
      INITIAL=6.E7
      LAST=12.E9
      XINC=(LAST-INITIAL)/(NSTEP-1)
        PI=3.14159265
C
C     THE SUMMATION IS FROM -N TO +N
C     COEFF - THE ARRAY OF COEFFICIENTS FOR iTH TERM IN THE SUMMATION
C             WHERE THE INDEX -N IS RESPRESENTED BY 1
C                             +N IS RESPRESENTED BY 2*N +1
C     T - ASSUMED CUTOFF TIME= 1/(2*SAMPLING FREQUENCY)
C
C     DATA STRUCTURE OF FNAME1
C     N       :NUMBER OF POINTS IN THE FILE (*)
C     FMIN    :SMALLEST FREQUENCY USED(*)
C     FREQ    :SAMPLING FREQUENCY (*)
C     COEFF(I):(REAL.IMAGINARY) (*)
C
      TYPE *,'TYPE FILE NAME CONTAINING COMPLEX COEFFICENTS'
      ACCEPT 2.FNAME1
2     FORMAT (A10)
      OPEN (UNIT=8,NAME=FNAME1.TYPE='OLD')
      READ(8,*) N
      READ(8,*) FMIN
      READ(8.*) FREQ
      T=1./(2.*FREQ)
      IF (N.GT.NUM) GO TO 8888
```

```
      NDIM=2*N +1
C
C     INITIALIZATION
C
      DO 5 I=1,MSIZE
            COEFF(I)=(0.,0.)
5     CONTINUE
      READ(8,*) COEFF(NUM)
      DO 10 I=1,N-1
            READ(8,*) COEFF(NUM+I+1)
C
C     FILL IN COMPLEX CONJUGATE FOR THE NEGATIVE FREQUENCY
C
            COEFF(NUM+1-I)=CONJG(COEFF(NUM+I+1))
10    CONTINUE
      CLOSE (UNIT=8,DISP='SAVE')
      NZERO=0
C
C     SET ANY FREQUENCY INTERPOLATION SPECIFIED BELOW FMIN TO ZERO
C     NZERO: NUMBER OF ZEROS TO BE PLACED
C
      IF (FMIN.LE.FREQ) GO TO 127
      NZERO=INT(FMIN/XINC)
      DO 123 I=1,NZERO
            DATA(I)=(0.,0.)
123   CONTINUE
C
C     STARTS THE INTERPOLATION
C     (THE SUMMATION IS PERFORMED BY FUNCTION SUM1D)
C
127   ILAST=INT((N-1)*FREQ/XINC)
      IF (ILAST.LT.NSTEP) INDEX=ILAST
      IF (ILAST.GE.NSTEP) INDEX=NSTEP
      DO  20 I=NZERO +1,INDEX
            DATA(I) = SUM1D(((I-1)*XINC+INITIAL)*T*2.*PI,COEFF,MSIZE,NDIM)
20    CONTINUE
C
C     SET ANY SPECIFIED FREQUENCY INTERPOLATION HIGHER THAN FMIN+N*FREQ
C     TO ZERO
C
      DO 133 I=INDEX +1,NSTEP
            DATA(I)=(0.,0.)
133   CONTINUE
C
C     PLOT AND OPTIONAL WRITE INTO A FILE: FNAME2
C
      CALL JPLOT(DATA,NSTEP,INITIAL,XINC,0,NSTEP-1)
23    WRITE(6,*) 'WRITE THE INTERPOLATED DATA INTO A FILE?'
      WRITE(6,*) '1)YES, IN AMPLITUDE AND PHASE(RADIAN) FORM'
      WRITE(6,*) '2)YES, IN REAL AND IMAGINARY FORM'
      WRITE(6,*) '3)NO, FORGET IT!'
```

```
      ACCEPT 25,COM1
25    FORMAT(A1)
      IF (COM1.EQ.'3') GO TO 9999
      IF ((COM1.NE.'1').AND.(COM1.NE.'2')) GO TO 23
      WRITE(6,*) 'STORAGE FILE NAME:'
      ACCEPT 30,FNAME2
30    FORMAT(A10)
      IF (COM1.EQ.'2') GO TO 35
C
C     CONVERT TO AMPLITUDE(LINEAR)  AND PHASE(RADIAN)
C
      DO 33 I=1,NSTEP
            XREAL=CABS(DATA(I))
            XIMAG=ATAN2(AIMAG(DATA(I)),REAL(DATA(I)))
            DATA(I)=CMPLX(XREAL,XIMAG)
33    CONTINUE
C
C     OUTPUT TO FNAME2
C     STRUCTURE OF FNAME2:
C     DATA(I):FREE COMPLEX FORMAT(*)
C
35    OPEN(UNIT=8,NAME=FNAME2,TYPE='NEW')
      DO 40 I=1,NSTEP
            WRITE(8,*) DATA(I)
40    CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
      GO TO 9999
8888  WRITE(6,*) 'ERROR:NOT ENOUGH SPACE SPECIFIED IN THE PROGRAM'
9999  STOP
      END
```

```
C
C     SUBROUTINE:    JPLOT
C
C     THIS SUBROUTINE WILL DO RECTANGULAR PLOT ON A COMPLEX ARRAY:'DATA'
C     FOR REAL AND IMAGINARY PLOT OR MAGNITUDE(LINEAR) AND PHASE(RADIAN)
C      PLOT.
C             DATA    :ARRAY NAME
C             NPOINT  :THE ARRAY SIZE
C             FMIN    :SMALLEST ELEMENT OF THE ABSCISSOR
C             FINCR   :THE INCRMENT SIZE
C             NSTART  :THE START PLOTTING INDEX
C             NLAST   :THE LAST PLOTTING INDEX
C     IF (NLAST.GT.NPOINT) THE LAST POINT PLOTTED IS NPOINT
C
C     THIS REQUIRES THE SUPPORT OF 'PLOTLIB.
C
      SUBROUTINE JPLOT(DATA,NPOINT,FMIN.FINCR.NSTART,NLAST)
      CHARACTER COM
      COMPLEX DATA(NPOINT)
      DIMENSION YAXIS1(9000),YAXIS2(9000),XAXIS(9000)
      PI=3.14159265
      MSIZE=9000
C
C     INITIALIZATION
C
      DO 3 I=1.MSIZE
             YAXIS1(I)=0.
             YAXIS2(I)=0.
             XAXIS(I)=0.
3     CONTINUE
5     WRITE(6,*) 'DO YOU WANT REAL AND IMAGINARY PLOT?Y/N'
      ACCEPT 6, COM
6     FORMAT(A1)
      IF (NLAST.GT.NPOINT) NLAST=NPOINT
      NP=NLAST-NSTART+1
      IF (COM.EQ.'N') GO TO 15
      IF (COM.NE.'Y') GO TO 5
C
C     REAL AND IMAGINARY PREPARATION
C
      DO  10 I=1.NP
             YAXIS1(I)=REAL(DATA(I+NSTART))
             YAXIS2(I)=AIMAG(DATA(I+NSTART))
             XAXIS(I)=(I-1+NSTART)*FINCR + FMIN
10    CONTINUE
      GO TO 25
C
C     GENERATE AMPLITUDE AND PHASE
C
15    DO 20 I=1.NP
             XAXIS(I)=(I+NSTART-1)*FINCR +FMIN
```

```
      YAXIS1(I)=CABS(DATA(I+NSTART))
      IF(REAL(DATA(I+NSTART)).EQ.0.) GO TO 18
      YAXIS2(I)=ATAN2(AIMAG(DATA(I+NSTART)),REAL(DATA(I+NSTART)))
      GO TO 20
18    IF (AIMAG(DATA(I+NSTART)).EQ.0.) YAXIS2(I)=0.
      IF (AIMAG(DATA(I+NSTART)).LT.0.) YAXIS2(I)=-PI/2
      IF (AIMAG(DATA(I+NSTART)).GT.0.) YAXIS2(I)=PI/2
20    CONTINUE
25    CALL PLTPKG(XAXIS.YAXIS1.9000.NP,1.0.1)
      CALL PLTPKG(XAXIS,YAXIS2,9000,NP,1,0.1)
      RETURN
      END
```

```
C
C     COMPLEX FUNCTION :      SUM1D
C
C     THIS SUBROUTINE WILL DO SUMMATION
C     X - NEW PARAMETER
C     Y - COEFFICIENT ARRAY
C     MSIZE-SIZE OF THE COEFFICIENT ARRAY
C     N - NUMBER OF COEFFICIENT ARRAY ELEMENTS THAT HAS NON-ZERO VALUES
C     NOTE:BOTHE MSIZE AND N ARE ODD NUMBERS
C
      COMPLEX FUNCTION SUM1D(X,Y,MSIZE,N)
      EXTERNAL SINC
      COMPLEX Y(MSIZE)
      PI=3.14159265
      MED=MSIZE/2 +1
      SUM1D=Y(MED)*SINC(X)
      DO 5  I=1,N/2
         SUM1D=SUM1D+Y(MSIZE/2-I+1)*SINC(X+I*PI)+Y(MSIZE/2+I+1)*SINC(X-I*PI)
5     CONTINUE
      RETURN
      END
```

```
C
C     FUNCTION:      SINC
C
C     THIS SUBROUTINE WILL CALCULATE SIN(X)/X
C     WHERE X IS ASSUMED TO BE IN RADIANS
C
      FUNCTION SINC(X)
      IF (X.EQ.0) GO TO 5
      SINC=(SIN(X))/X
      GO TO 10
5     SINC=1.
10    RETURN
      END
```

```
C
C     PROGRAM :        PTGRID
C
C     THIS PROGRAM WILL GENERATE ALL THE ANGLES AND FREQUENCIES
C     FOR THE GRID IN THE FREQUENCY DOMAIN.  THE DATA ARE
C     ARRANGED IN ORDER FROM SMALLEST ANGLE TO THE LARGEST ANGLE.
C     (0 TO 2*PI) THE SMALLEST FREQUENCY
C     RADIUS IS STORED.  SO AT THE TIME OF MEASUREMENT, ONLY
C     MULTIPLICATION OF THE RADIUS IS NECESSARY.
C
C     THE GRID POINTS ARE WRITE INTO FILE:'OUT.DAT'
C     THE DATA FORMAT OF 'OUT.DAT':
C     FMIN.FMAX          :FREQUENCY RANGE SPECIFIED(2E15.8)
C     FDELTA             :SAMPLING FREQUENCY(E15.8)
C     NPOINT             :NUMBER OF POINTS IN THE FILE(I8)
C     ISO,RADIUS         :ISO=N. NON-ISOROPIC SAMPLING IS USED(A2)
C                        :ISO=T, ISOTROPIC SAMPLING IS USED
C                        :     RADIUS:RADIUS OF THE ISOTROPIC CELL IN MM(E15.8)
C     V1.V2              :VECTORS USED TO DEFINE THE SAMPLING LATTICE(4E15.8)
C     U1.U2              :VECTORS USED TO DEFINE THE PERIODIC LATTICE(4E15.8)
C     DATA(*,1),DATA(*,2): DATA(4E15.8)
C     DATA(*,1) IS THE MEASUREMENT ANGLE AND FREQUENCY ARRAY
C     DATA(*,2) IS THE INDEX SPECIFYING HOW MANY UNITS OF
C     V1.V2  ARE USED
C
C     WARNING: SIZE OF DATA IS (10000,2)
C
C     LINK PTGRID,GEN1.GEN2.SEARCH.INSERT,PUSH.NORMAL.'SSP
C
      COMPLEX V1.V2.V3.V4.U1.U2.DATA(10000.2)
      DIMENSION WORK1(2),WORK2(2)
      REAL MAG1.MAG2.MAX1,MARGIN.MATRIX(4)
      CHARACTER  ISO
      MSIZE=10000
      PI=3.14159265
C
C     START WORKING
C
2     WRITE (6.*) ' LOWEST AND HIGHEST OPERATING FREQUENCY IN HERTZ'
      ACCEPT *, FMIN,FMAX
      IF (FMIN.GE.FMAX) GO TO 2
3     WRITE (6.*) ' DO YOU WANT TO DO ISOTROPIC SAMPLING ? T/F '
      ACCEPT 4.ISO
4     FORMAT (A1)
      IF (ISO.EQ.'F') GO TO 5
      IF (ISO.NE.'T') GO TO 3
      WRITE (6.*) ' DIAMETER OF NORMALIZATION IN MM?'
      ACCEPT *,RADIUS
C
C     DEFINITION OF ISOTROPIC VECTORS
C
```

```
      X1=(SQRT(3.))/2.
      Y1=-0.5
      X2=0.
      Y2=1.
      GO TO 6
C
C     CASE OF PARALLELPIPEDIC CONFINEMENT
C
5     WRITE (6,*) 'THE OBJECT IS ASSUMED TO BE CONFINED TO A PARALLELOGRAM!'
      WRITE (6,*) ' NORMALIZATION FACTOR IN MM'
      ACCEPT *,RADIUS
      WRITE (6,*) ' VECTOR1 IN X,Y :'
      ACCEPT *, X1,Y1
      WRITE (6,*) ' VECTOR2 IN X,Y :'
      ACCEPT *, X2,Y2
6     WRITE (6,*) ' SAFETY MARGIN FACTOR ON THE TIME WAVEFORM'
      ACCEPT *,MARGIN
      U1=CMPLX(X1,Y1)
      U2=CMPLX(X2,Y2)
      MATRIX(1)=(X1)
      MATRIX(2)=(Y1)
      MATRIX(3)=(X2)
      MATRIX(4)=(Y2)
C
C     MINV WILL DO MATRIX INVERSION.  THIS SUBROUTINE IS IN SSP
C
      CALL MINV(MATRIX,2,D,WORK1,WORK2)
C
C     NOTE THAT V1 AND V2 IS TAKEN FROM THE TRANSPOSE
C     OF THE INVERSE OF MATRIX.  THIS IS TO BE CONSISTENCE WITH
C     DOT PRODUCT OF U(I) AND V(J) = DELTA(IJ)
C
C     U IS IN SPATIAL DOMAIN
C     V IS IN FREQUENCY DOMAIN
C     ALSO V1=(X-COMPONENT,Y-COMPONENT)
C
      V1=CMPLX(MATRIX(1),MATRIX(3))
      V2=CMPLX(MATRIX(2),MATRIX(4))
      MAG1=CABS(V1)
      PHASE1=ATAN2(AIMAG(V1),REAL(V1))
      MAG2=CABS(V2)
      PHASE2=ATAN2(AIMAG(V2),REAL(V2))
      THETA=ABS(PHASE1-PHASE2)
C
C     PHASES ARE NORMALIZED TO THE RANGE 0:2*PI
C
      CALL NORMAL(PHASE1)
      CALL NORMAL(PHASE2)
C
C     SFACTOR: FREQUENY SAMPLING FACTOR
C
```

```
      SFACTOR=(3.E11)/(RADIUS*MARGIN)
      IF ((SFACTOR.LE.FMIN).OR.(FMIN.EQ.0.)) FDELTA=SFACTOR
      IF ((SFACTOR.GT.FMIN).AND.(FMIN.NE.0.)) FDELTA=FMIN
      RADIUS=1/(FDELTA*2.)
      MAX1=(FMAX/FDELTA)
C
C     STORE THE FIRST TWO VECTORS INTO THE ARRAY 'DATA'
C
      IF (PHASE2.LT.PHASE1) GO TO 11
      DATA(1,1)=CMPLX(PHASE1,MAG1)
      DATA(1,2)=CMPLX(1,0)
      DATA(2,1)=CMPLX(PHASE2,MAG2)
      DATA(2,2)=CMPLX(0,1)
      GO TO 12
11    DATA(1,1)=CMPLX(PHASE2,MAG2)
      DATA(1,2)=CMPLX(0,1)
      DATA(2,1)=CMPLX(PHASE1,MAG1)
      DATA(2,2)=CMPLX(1,0)
12    NPOINT=2
C
C      TO GENERATE THE FIRST QUADRANT POINTS
C     POINTS ARE GENERATED AND STORED IN SUBROUTINE GEN1 AND GEN2
C
      IF (THETA.LE.PI/2.)
     &                CALL GEN1(V1,V2,MAX1,MAX1,DATA,MSIZE,NPOINT,1,1)
      IF (THETA.GT.PI/2.)
     &                CALL GEN2(V1,V2,MAX1,MAX1,DATA,MSIZE,NPOINT,1,1)
      IF (NPOINT.GT.MSIZE) GO TO 888
      V3=-V1
      PHASE3 = ATAN2(AIMAG(V3),REAL(V3))
      CALL NORMAL(PHASE3)
      V3=CMPLX(PHASE3,CABS(V3))
C
C     SEARCH FOR LOCATION :LOC TO PLACE THE POINT
C
      LOC=SEARCH(V3,DATA,MSIZE,NPOINT)
C
C     INSERT THE POINT INTO THE PROPER LOCATION
C
      CALL INSERT(LOC,NPOINT,DATA,MSIZE,V3,-1.,0.)
C
C      TO GENERATE THE 2ND QUADRANT POINTS
C
      IF (THETA.LE.PI/2.)
     &                CALL GEN2(V1,V2,MAX1,MAX1,DATA,MSIZE,NPOINT,-1,1)
      IF (THETA.GT.PI/2.)
     &                CALL GEN1(V1,V2,MAX1,MAX1,DATA,MSIZE,NPOINT,-1,1)
      IF (NPOINT.GT.MSIZE) GO TO 888
      V4=-V2
      PHASE4 = ATAN2(AIMAG(V4),REAL(V4))
      CALL NORMAL(PHASE4)
```

```
      V4=CMPLX(PHASE4.CABS(V4))
      LOC=SEARCH(V4,DATA,MSIZE,NPOINT)
      CALL INSERT(LOC,NPOINT,DATA,MSIZE,V4.0.,-1.)
C
C     NOW TO GENERATE THE OTHER TWO QUADRANTS
C
      IF (THETA.LE.PI/2.)
     &              CALL GEN2(V1.V2.MAX1.MAX1.DATA,MSIZE,NPOINT,1,-1)
      IF (THETA.GT.PI/2.)
     &              CALL GEN1(V1,V2.MAX1,MAX1,DATA,MSIZE,NPOINT,1,-1)
      IF (NPOINT.GT.MSIZE) GO TO 888
      IF (THETA.LE.PI/2.)
     &              CALL GEN1(V1,V2.MAX1,MAX1.DATA,MSIZE.NPOINT,-1,-1)
      IF (THETA.GT.PI/2.)
     &              CALL GEN2(V1,V2.MAX1,MAX1.DATA,MSIZE,NPOINT,-1,-1)
      IF (NPOINT.GT.MSIZE) GO TO 888
C
C     OUTPUT GRID POINTS INTO FILE:'OUT.DAT'
C
      OPEN(UNIT=8,NAME='OUT',TYPE='NEW')
      WRITE(8.301) FMIN.FMAX
301   FORMAT(2E15.8)
      WRITE(8,305) FDELTA
305   FORMAT(E15.8)
      WRITE(8,310) NPOINT
310   FORMAT(I8)
      WRITE(8,315), ISO,RADIUS
315   FORMAT(A2.E15.8)
      WRITE(8.320) V1.V2
320   FORMAT(2E15.8,2E15.8)
      WRITE(8.320) U1.U2
      DO 20 J=1.NPOINT
              WRITE(8,320) DATA(J,1),DATA(J,2)
20    CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
      GO TO 999
888   WRITE(6.*) 'ERROR:SPECIFIED ARRAY SIZE TOO SMALL!'
999   STOP
      END
```

```
C
C     SUBROUTINE:     GEN1
C
C     THIS SUBROUTINE WILL GENERATE ALL THE GRID POINTS WITHIN THE TWO
C     VECTORS V1.V2 ON THE TWO DIMENSIONAL PLANE.  THAT IS IF THE ANGLE
C     IS ACUTE.
C
C     V1       :FIRST VECTOR
C     V2       :SECOND VECTOR
C     MAX      :MAXIMUM FREQUENCY IN EACH VECTOR DIRECTION
C     AMAX     :ABSOLUTION MAXIMUM FREQUENCY
C     DFILE    :DATA ARRAY FOR STORAGE
C     MSIZE    :SIZE OF DFILE
C     M        :NUMBER OF POINTS GENERATED
C     N1SIGN   :SIGN OF N1  (N1*V1)
C     N2SIGN   :SIGN OF N2  (N2*V2)
C
C     THIS REQUIRES THE SUPPORT OF SEARCH.NORMAL.INSERT
      SUBROUTINE GEN1(V1.V2.MAX,AMAX,DFILE,MSIZE,M,N1SIGN,N2SIGN)
      EXTERNAL SEARCH
      COMPLEX DFILE(MSIZE.2),VECTOR,V1.V2
      REAL MAG,MAX
C
C     CALCULATE THE NUMBER OF UNITS IN EACH VECTOR DIRECTIONS
C
      N1=INT(MAX/(CABS(V1))) + 1
      N2=INT(MAX/(CABS(V2))) + 1
      DO 5 I=1.N2
              DO 10 J=1.N1
                      VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
                      MAG=CABS(VECTOR)
                      IF (MAG.GT.AMAX) GO TO 5
                      PHASE=ATAN2(AIMAG(VECTOR),REAL(VECTOR))
                      CALL NORMAL(PHASE)
                      VECTOR=CMPLX(PHASE,MAG)
C
C     SEARCH THE LOCATION
C
                      LOC=SEARCH(VECTOR,DFILE,MSIZE,M)
                      RJ=FLOAT(J)
                      RI=FLOAT(I)
C
C     INSERT THE DATA
C
                      IF ((N1SIGN.LT.0).AND.(N2SIGN.LT.0))
     &                          CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ.-RI)
                      IF ((N1SIGN.LT.0).AND.(N2SIGN.GT.0))
     &                          CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
                      IF ((N1SIGN.GT.0).AND.(N2SIGN.LT.0))
     &                          CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
                      IF ((N1SIGN.GT.0).AND.(N2SIGN.GT.0))
```

```
     &                    CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)
10            CONTINUE
5     CONTINUE
      RETURN
      END
```

```
C
C     SUBROUTINE:    GEN2
C
C     THIS SUBROUTINE WILL GENERATE POINTS BETWEEN VECTORS V1.V2 IF
C     THE ANGLE BETWEEN THEM IS MORE THAN 90 DEGREES BUT
C     LESS THAN 180 DEGREES
C
C
C     V1       :FIRST VECTOR
C     V2       :SECOND VECTOR
C     MAX      :MAXIMUM FREQUENCY IN EACH VECTOR DIRECTION
C     AMAX     :ABSOLUTION MAXIMUM FREQUENCY
C     DFILE    :DATA ARRAY FOR STORAGE
C     MSIZE    :SIZE OF DFILE
C     M        :NUMBER OF POINTS GENERATED
C     N1SIGN   :SIGN OF N1 (N1*V1)
C     N2SIGN   :SIGN OF N2 (N2*V2)
C
C     THIS REQUIRES THE SUPPORT OF GEN1,INSERT,SEARCH
C
      SUBROUTINE GEN2(V1.V2.MAX.AMAX.DFILE,MSIZE,M,N1SIGN,N2SIGN)
      EXTERNAL SEARCH
      COMPLEX DFILE(MSIZE,2),VECTOR,V1,V2
      REAL MAG,MAX
      LOGICAL SAME
      PI= 3.14159265
      THETA=ABS(ATAN2(AIMAG(V1),REAL(V1)) -ATAN2(AIMAG(V2),REAL(V2)))
      XMAX=MAX/SIN(THETA)
      CALL GEN1(V1.V2.XMAX,AMAX,DFILE.MSIZE.M,N1SIGN.N2SIGN)
      IF (THETA.EO.PI/2.) GO TO 555
      N1CORRECT=0
      N2CORRECT=0
      IF(AMOD(XMAX.CABS(V1)).GT.0.) N1CORRECT=1
      IF(AMOD(XMAX.CABS(V2)).GT.0.) N2CORRECT=1
      N1=INT(XMAX/(CABS(V1))) + N1CORRECT
      N2=INT(XMAX/(CABS(V2))) + N2CORRECT
      I2=INT(MAX/(CABS(V2)))
C
C     TO CHECK OUT WHICH REGION ARE THE VECTORS IN
C
      VECTOR=V1*N1SIGN + V2*N2SIGN
      IF((REAL(VECTOR)*AIMAG(VECTOR)).GT.0) SAME=.TRUE.
      IF ((REAL(VECTOR)*AIMAG(VECTOR)).LT.0) SAME=.FALSE.
      IF ((REAL(VECTOR)*AIMAG(VECTOR)).NE.0) GO TO 3
      VECTOR=2.*V1*N1SIGN + V2*N2SIGN
      IF((REAL(VECTOR)*AIMAG(VECTOR)).GT.0) SAME=.TRUE.
      IF ((REAL(VECTOR)*AIMAG(VECTOR)).LT.0) SAME=.FALSE.
3     CHECK=0.
      DO 5 J=1.N1
              DO 10 I=I2.N2
                      IF(CHECK.GE.2.) GO TO 50
```

```
                         VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
                         MAG=CABS(VECTOR)
C
C     MAKE SURE THAT CHECK IS INCREMENTED PROPERLY
C
                         IF ((NOT SAME).AND.((REAL(VECTOR)*AIMAG(VECTOR)).GT.0))
     &                                CHECK=CHECK + 1.
                         IF ((SAME).AND.((REAL(VECTOR)*AIMAG(VECTOR)).LT.0))
     &                                CHECK=CHECK + 1
C
C     MAKE SURE THE POINT GENERATED IS INSIDE THE CIRCLE
C
                         IF (MAG.GT.AMAX) GO TO 5
C
C     MAKE SURE THAT CHECK IS DECREMENTED PROPERLY
C
                         IF ((NOT SAME).AND.((REAL(VECTOR)*AIMAG(VECTOR)).GT.0))
     &                                CHECK=CHECK - 1.
                         IF ((SAME).AND.((REAL(VECTOR)*AIMAG(VECTOR)).LT.0))
     &                                CHECK=CHECK - 1.
                         PHASE=ATAN2(AIMAG(VECTOR),REAL(VECTOR))
                         CALL NORMAL(PHASE)
                         VECTOR=CMPLX(PHASE,MAG)
                         LOC=SEARCH(VECTOR,DFILE,MSIZE,M)
                         RJ=FLOAT(J)
                         RI=FLOAT(I)
                         IF ((N1SIGN.LT.0).AND.(N2SIGN.LT.0))
     &                           CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
                         IF ((N1SIGN.LT.0).AND.(N2SIGN.GT.0))
     &                           CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
                         IF ((N1SIGN.GT.0).AND.(N2SIGN.LT.0))
     &                           CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
                         IF ((N1SIGN.GT.0).AND.(N2SIGN.GT.0))
     &                           CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)
10               CONTINUE
5     CONTINUE
50    I1=J-2
      DO 15 I=I2,N2
              DO 20 J=I1,N1
                         VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
                         MAG=CABS(VECTOR)
                         IF (MAG.GT.AMAX) GO TO 15
                         PHASE=ATAN2(AIMAG(VECTOR),REAL(VECTOR))
                         CALL NORMAL(PHASE)
                         VECTOR=CMPLX(PHASE,MAG)
                         LOC=SEARCH(VECTOR,DFILE,MSIZE,M)
                         RJ=FLOAT(J)
                         RI=FLOAT(I)
                         IF ((N1SIGN.LT.0).AND.(N2SIGN.LT.0))
     &                           CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
                         IF ((N1SIGN.LT.0).AND.(N2SIGN.GT.0))
```

```
     &                        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
                  IF ((N1SIGN.GT.0).AND.(N2SIGN.LT.0))
     &                        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
                  IF ((N1SIGN.GT.0).AND.(N2SIGN.GT.0))
     &                        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)
20          CONTINUE
15    CONTINUE
555   RETURN
      END
```

```
C
C     FUNCTION:      SEARCH
C
C     THIS FUNCTION WILL SEARCH THE LOCATION WHERE THE ELEMENT
C     FITS INTO A DATA ARRAY.  THE ARRAY ASSUMES AT LEAST TWO
C     MEMBERS.  THE ELEMENT MAY HAVE 3 POSSIBILITIES:-
C             1) THE REAL PART IS THE SAME IN THE RETURNED LOCATION
C             2) THE REAL PART IS THE SAME IN THE RETURNED LOCATION +1
C             3) THE REAL PART IS BETWEEN THE ABOVE TWO
C
C     THIS FUNCTION EMPLOYES BINARY SEARCH TECHNIQUE TO LOCATE
C
C     ELEMENT :COMPLEX ELEMENT TO BE PLACED WITH PRIORITY OF THE
C              REAL PART OVER THE IMAGINARY PART
C     DFILE   :ARRAY FILE TO BE SEARCHED AND INSERTED
C     MSIZE   :SIZE OF DFILE
C     LAST    :NUMBER OF ELEMENT IN DFILE
C
      FUNCTION SEARCH(ELEMENT,DFILE,MSIZE,LAST)
      COMPLEX DFILE(MSIZE,2),ELEMENT
      LOC2=LAST
      LOC1=1
      IF (REAL(ELEMENT).LT.REAL(DFILE(LOC1,1))) GO TO 147
      IF (REAL(ELEMENT).GT.REAL(DFILE(LOC2.1))) GO TO 140
120   LOCM = (LOC1 +LOC2)/2
      IF(LOCM.EQ.LOC1) GO TO 150
      IF (REAL(ELEMENT).LT.REAL(DFILE(LOCM,1))) GO TO 125
      IF (REAL(ELEMENT).GT.REAL(DFILE(LOCM,1))) GO TO 130
      GO TO 145
125   LOC2 =LOCM
      GO TO 120
130   LOC1=LOCM
      GO TO 120
140   SEARCH = LOC2
      GO TO 155
145   SEARCH = LOCM
      GO TO 155
147   SEARCH = LOC1-1
      GO TO 155
150   SEARCH = LOC1
155   RETURN
      END
```

```
C
C     SUBROUTINE:     INSERT
C
C     THIS SUBROUTINE WILL INSERT THE ELEMENT INTO THE DFILE
C     IF THE ELEMENT HAS A DIFFERENT PHASE OR A SMALLER
C     MAGNITUDE.  THE PHASE HAS A SENSITIVITY SPECIFIED BY THE ERROR.
C
C     NSTART  :LOCATION OF THE ELEMENT
C     LAST    :NUMBER OF ELEMENTS IN DFILE
C     DFILE   :ARRAY FILE TO BE INSERTED
C     MSIZE   :DIMENSION OF DFILE
C     ELEMENT :COMPLEX ELEMENT TO BE INSERTED
C     V1INDEX :NUMBER OF V1 USED
C     V2INDEX :NUMBER OF V2 USED
C
      SUBROUTINE INSERT(NSTART,LAST,DFILE,MSIZE,ELEMENT,V1INDEX,V2INDEX)
      COMPLEX DFILE(MSIZE,2),ELEMENT
      PI=3.14159265
      ERROR= 0.01*PI/180.
      IF (NSTART.NE.LAST) GO TO 200
      IF(ABS(REAL(ELEMENT)-REAL(DFILE(LAST,1))).LT.ERROR) GO TO 290
      LAST = LAST +1
      DFILE(LAST,1)=ELEMENT
      DFILE(LAST,2)=CMPLX(V1INDEX,V2INDEX)
      GO TO 295
200   IF (NSTART.EQ.0) GO TO 202
      IF (ABS(REAL(ELEMENT)-REAL(DFILE(NSTART,1))).LT.ERROR) GO TO 285
      IF (ABS(REAL(DFILE(NSTART+1,1))-REAL(ELEMENT)).LT.ERROR) GO TO 280
202   CALL PUSH(NSTART+1,LAST,DFILE,MSIZE)
      DFILE(NSTART+1,1)=ELEMENT
      DFILE(NSTART+1,2)=CMPLX(V1INDEX,V2INDEX)
      GO TO 295
280   IF(AIMAG(ELEMENT).GE.AIMAG(DFILE(NSTART+1,1))) GO TO 295
      DFILE(NSTART+1,1) = ELEMENT
      DFILE(NSTART+1,2)=CMPLX(V1INDEX,V2INDEX)
      GO TO 295
285   IF(AIMAG(ELEMENT).GE.AIMAG(DFILE(NSTART,1))) GO TO 295
      DFILE(NSTART,1)=ELEMENT
      DFILE(NSTART,2)=CMPLX(V1INDEX,V2INDEX)
      GO TO 295
290   IF(AIMAG(ELEMENT).GE.AIMAG(DFILE(LAST,1))) GO TO 295
      DFILE(LAST,1)=ELEMENT
      DFILE(LAST,2)=CMPLX(V1INDEX,V2INDEX)
295   RETURN
      END
```

```
C
C     SUBROUTINE:     PUSH
C
C     THIS SUBROUTINE WILL PUSH ALL THE DATA IN 'DFILE' BY 1 POSITION
C     THE OPERATION IS SPECIFIED BY NSTART1 -THE BEGINNING OF PUSH
C                                   LAST1   -THE LAST LOCATION OF DFILE
C
      SUBROUTINE PUSH(NSTART1.LAST1.DFILE.MSIZE)
      COMPLEX DFILE(MSIZE,2),TEMP1,TEMP2
      N=LAST1- NSTART1 +1
      DO 5 I=1,N
              TEMP1=DFILE(LAST1+1-I,1)
              TEMP2=DFILE(LAST1+1-I,2)
              DFILE(LAST1+2-I,1)=TEMP1
              DFILE(LAST1+2-I,2)=TEMP2
5     CONTINUE
      LAST1 = LAST1 +1
      RETURN
      END
```

```
C
C     SUBROUTINE:     NORMAL
C
C     THIS SUBROUTINE WILL NORMALIZE A PHASE RANGE OF (-PI TO PI)
C     TO A RANGE OF (0 TO 2*PI).
C
C     ALPHA: ANGLE IN RADIAN TO BE NORMALIZED
C
      SUBROUTINE NORMAL(ALPHA)
      PI=3.14159265
      IF (ALPHA.LT.0) ALPHA=ALPHA + 2.*PI
      RETURN
      END
```

```
C
C     PROGRAM:        INTERPOL
C
C     THIS PROGRAM WILL INTERPOLATE FOR 3 OPTIONS.
C     FILE FNAME STORED THE GRID POINT INFORMATION GENERATED PREVIOUSLY.
C     THIS PROGRAM WILL REPEAT ITSELF, EXIT ONLY BY CONTROL C
C     THIS PROGRAM LINKS WITH 'PLOTLIB
C
C     THIS PROGRAM REQUIRES AN INPUT OF A MEASUREMENT FILE :FNAME
C     THE STRUCTURE OF FNAME:
C     FMIN.FMAX -MAXIMUM AND MINIMUM FREQUENCY USED IN HZ(2E15.8)
C     FDELTA    -NORMALIZATION FACTOR IN FREQUENCY(1E15.8)
C     NPOINT    -NUMBER OF POINTS IN THIS FILE(I8)
C     ISO,RADIUS-ISO='T' IF ISOTROPIC SAMPLING WAS DONE(A2)
C               -ISO='F' IF ISOTROPIC SAMPLING WAS NOT DONE
C               -RADIUS IS THE ISOTROPIC CIRCLE RADIUS USED IN SECOND(E15.8)
C     V1,V2     -VECTOR1 AND VECTOR2 IN FREQUENCY DOMAIN(4E15.8)
C     U1,U2     -VECTOR1 AND VECTOR2 IN TIME/SPACE DOMAIN(4E15.8)
C     *,*,*,*-REAL AND IMAG MEASUREMENT;N1,N2 FOR THE MEASUREMENT(4E15.8)
C     (MAKE SURE ALL THE ABOVE ARGURMENTS ARE SEPARATED BY COMMAS IN FNAME!)
C
C     N1*V1,N2*V2 SPECIFIES THE SAMPLING LATTICE
C
C     LINK INTERPOL,WFILE,SUM,CONST1,CONST2,FUNC1,FUNC2,FUNC3,SINC,-
C            RPLOT,ELIM,'PLOTLIB
C
      COMPLEX SUM,DATA(360),V1,V2,U1,U2,FREQ,COEFF(10000,2),TEMP1
      CHARACTER ISO
      CHARACTER*10 FNAME
      MSIZE=10000
C
C     NDATA    :SPECIFIES THE NUMBER OF ANGULAR POINTS
C     NUM      :SPECIFIES THE NUMBER OF FREQUENCY POINTS
C
      NDATA=360
      NUM=201
      PI=3.14159265
      WRITE(6,*) ' FILE WHICH HAS GRID POINT MEASUREMENT?'
      ACCEPT 50, FNAME
50    FORMAT(A10)
      OPEN(UNIT=9,NAME=FNAME,TYPE='OLD')
C
C     READ IN NECESSARY DATA
C
      READ(9,550) FMIN,FMAX
550   FORMAT(2E15.8)
      READ(9,551) FDELTA
551   FORMAT(E15.8)
      READ(9,552) NPOINT
552   FORMAT(I8)
      IF (NPOINT.GT.MSIZE) GO TO 8888
```

```
      READ(9,553) ISO,RADIUS
553   FORMAT(A2.E15.8)
      READ(9.554) V1.V2
554   FORMAT(2E15.8,2E15.8)
      READ(9,554) U1,U2
      V1=2.*PI*FDELTA*V1
      V2=2.*PI*FDELTA*V2
      U1=U1/(FDELTA)
      U2=U2/FDELTA
C
C     OPTIONS TO GUARD AGAINST MEASUREMENT DATA IN FORMAT
C     OTHER THAN REAL AND IMAGINERY
C
570   WRITE(6,*) 'DATA IN WHICH FORMAT:1) REAL AND IMAG'
      WRITE(6,*) '                     2) AMP(LINEAR) AND PHASE(RADIAN)'
      WRITE(6,*) '                     3) AMP(DB)     AND PHASE(RADIAN)'
      WRITE(6,*) '                     4) AMP(DB)     AND PHASE(DEGREE)'
      ACCEPT *,NCOM1
      IF ((NCOM1.GT.4).OR.(NCOM1.LT.1)) GO TO 570
      DO 560 I=1.NPOINT
              READ(9.554) XREAL.XIMAG,COEFF(I,2)
              IF(NCOM1.EQ.1) GO TO 580
              AMP=XREAL
              PHASE=XIMAG
              IF((NCOM1.EQ.3).OR.(NCOM1.EQ.4)) AMP=10**(AMP/20)
              IF(NCOM1.EQ.4) PHASE=PHASE*PI/180.
              XREAL=AMP*COS(PHASE)
              XIMAG=AMP*SIN(PHASE)
580           COEFF(I,1)=CMPLX(XREAL.XIMAG)
560   CONTINUE
      CLOSE(UNIT=9.DISP='SAVE')
C
C     ASK FOR THE TYPE OF INTERPOLATION
C
5     WRITE(6,*) ' 3 CHOICES :1) INTERPOLATE FOR 1 ASPECT ANGLE AND OUTPUT'
      WRITE(6,*) '               201 FREQUENCY POINTS.'
      WRITE(6,*) '            2) INTERPOLATE FOR 1 FREQUENCY AND OUTPUT  '
      WRITE(6,*) '               360 DEGREE POINTS.'
      WRITE(6,*) '            3) INTERPOLATE INTO 2-D SQUARE GRID'
      ACCEPT *, NCOM
C
C     THE BRANCHING OF DECISION
C
      IF (NCOM.EQ.2) GO TO 200
      IF (NCOM.EQ.3) GO TO 300
      IF (NCOM.NE.1) GO TO 5
C
C     TO INTERPOLATE FOR 1 ASPECT ANGLE BUT 201 FREQUENCY POINTS
C
10    WRITE(6,*) 'FREQUENCY RANGE? LOW TO HIGH IN HZ'
      ACCEPT *, FMINS,FMAXS
```

```
      IF (FMINS.GE.FMAXS) GO TO 10
      WRITE(6,*) ' WHICH ASPECT ANGLE?(degree)'
      ACCEPT *, ANGLE
      ANGLE=ANGLE*PI/180.
      FINCR=(FMAXS-FMINS)/(NUM-1)
      DO 15 I=1,NUM
              WRITE(6,*) 'WORKING ON: ',I,' LAST ONE: ',NUM
              AFREQ=FMINS + FINCR*(I-1)
              IF (AFREQ.EQ.0.) GO TO 13
C
C     MAKE SURE THE INTERPOLATION IS DONE BETWEEN THE MEASURED FREQUENCY
C     RANGE
C
              IF (((AFREQ).GT.FMAX).OR.((AFREQ).LT.FMIN)) GO TO 13
              FREQ=CMPLX(AFREQ*COS(ANGLE),AFREQ*SIN(ANGLE))
              DATA(I)= SUM(FREQ,COEFF,MSIZE,V1,V2,U1,U2,ISO,NPOINT,RADIUS)
              GO TO 15
13            DATA(I)=(0.,0.)
15    CONTINUE
C
C     PLOT AND WRITE FILE
C
      CALL RPLOT(DATA,NUM,FMINS,FINCR)
      CALL WFILE(DATA,NDATA,NUM)
      GO TO 5
C
C     SECOND SECTION OF THE PROGRAM
C     TO INTERPOLATE FOR 1 FREQUENCY, BUT 360 DEGREE POINTS
C
200   WRITE(6,*) 'WHICH FREQUENCY?(HZ)'
      ACCEPT *, AFREQ
      DO 205 I=1,NDATA
              WRITE(6,*) 'WORKING ON: ',I,' LAST ONE: ',NDATA
              ANGLE=(I-1)*PI/180.
              IF (AFREQ.EQ.0.) GO TO 203
C
C     MAKE SURE THE INTERPOLATION IS DONE BETWEEN THE MEASURED FREQUENCY
C     RANGE
C
              IF (((AFREQ).GT.FMAX).OR.((AFREQ).LT.FMIN)) GO TO 203
              FREQ=CMPLX(AFREQ*COS(ANGLE),AFREQ*SIN(ANGLE))
              DATA(I)= SUM(FREQ,COEFF,MSIZE,V1,V2,U1,U2,ISO,NPOINT,RADIUS)
              GO TO 205
203           DATA(I)=(0.,0.)
205   CONTINUE
C
C     PLOT AND WRITE THE FILE
C     POLARP  :A SUBROUTINE IN 'PLOTLIB
C
      CALL POLARP(DATA,3.5,3.1,NDATA,1,1)
      CALL WFILE(DATA,NDATA,NDATA)
```

```
      GO TO 5
C
C     THIRD PART OF THE PROGRAM
C     TO INTERPOLATE INTO A 2-D SQUARE GRID
C     (DUE TO TIME CONSUMPTION, THIS IS NOT IMPLEMENTED.  BUT IT CAN
C      BE DONE SIMILARILY AS PART 1 AND 2)
C
300   WRITE(6,*) 'NOT READY YET!'
      GO TO 5
8888  WRITE(6,*) 'ERROR:THE SIZE OF INPUT FILE IS LARGER THAN SPECIFIED!'
      GO TO 5
      END
```

```
C
C     COMPLEX FUNCTION:        SUM
C
C     THIS FUNCTION WILL DO THE SUMMATION OF F(N1*V1+N2*V2)*G(W-N1*V1-N2*V2)
C     (EQUATION 2-9 IN THIS WRITE UP)
C     FOR ALL AVAILABLE VALUES OF N1 AND N2 IN THE FILE FNAME
C
C     FREQ    :COMPLEX (F1,F2); LOCATION IN THE FREQUENCY PLANE
C     COEFF   :COMPLEX ARRAY STORING THE SAMPLED VALUES
C     MSIZE   :SIZE OF COEFF
C     V1.V2   :VECTOR 1 AND 2 IN THE FREQUENCY DOMAIN
C     U1.U2   :VECTOR 1 AND 2 IN THE SPACE DOMAIN
C     ISO     :T=ISOTROPIC SAMPLING ,F=NON-ISOTROPIC SAMPLING(A1)
C     NPOINT  :NUMBER OF NON-ZERO ELEMENTS IN COEFF
C     RADIUS  :RADIUS OF THE ISOTROPIC  CELL
C
C     THIS REQUIRES THE SUPPORT OF CONST1.CONST2
C
      COMPLEX FUNCTION SUM(FREQ.COEFF,MSIZE,V1.V2.U1,U2.ISO,NPOINT,RADIUS)
      CHARACTER       ISO
      COMPLEX FREQ,V1,V2-U1.U2.W,COEFF(MSIZE,2)
      PI=3.14159265
      SUM=CMPLX(0.,0.)
C
C     CONVERSION OF FREQUENCY INTO RADIAN FREQUENCY
C
      FREQ=2.*PI*FREQ
      DO 5 I=1.NPOINT
             W=FREQ-REAL(COEFF(I,2))*V1-AIMAG(COEFF(I,2))*V2
             IF (ISO.EQ.'T') SUM=SUM + COEFF(I,1)*CONST1(RADIUS,W)
             IF (ISO.EQ.'F') SUM=SUM + COEFF(I,1)*CONST2(U1.U2.W)
5     CONTINUE
      RETURN
      END
```

```
C
C     FUNCTION        :CONST1
C
C     THIS FUNCTION IS FOR RECONSTRUCTION OF ISOTROPIC SAMPLING
C     CALCULATING:
C     (1/((R**2)*W1*(W1**2-3*W2**2)))*( 2*W1*COS(RW1/SQRT(3))*COS(R*W2)
C                                     -2*W1*COS(2*R*W1/SQRT(3))
C                                -2*SQRT(3)*W2*SIN(R*W1/SQRT(3))*SIN(R*W2))
C
C     THIS REQUIRES THE SUPPORT OF FUNC1,FUNC2.FUNC3,ELIM
C
C     R:CONSTANT
C     W:COMPLEX (W1.W2)
C
      FUNCTION CONST1(R,W)
      COMPLEX W
      DIMENSION AB(3,4)
      TOLER=1.
      W1=REAL(W)
      W2=AIMAG(W)
      C=SQRT(3.)
      D1=(W1-C*W2)*R
      D2=(W1+C*W2)*R
C
C     THE RECONSTRUCTION FUNCTION WILL BLOW UP WHEN 1)W1=0 OR
C                                                   2)W1**2=3*W2**2
C
      IF ((ABS(D1).GE.TOLER).AND.(ABS(D2).GE.TOLER)) GO TO 999
      IF ((D1.EQ.0.).OR.(D2.EQ.0.)) GO TO 888
C
C     THERE IS ONE MORE PROBLEM DURING IMPLEMENTATION:
C     THE COMPUTER'S UNDERFLOW PROBLEM.
C     THEREFORE FOR TOLER=1., THE FUNCTION IS INTERPOLATED
C     WITH THREE POINT MATCHING POLYNOMINAL.
C     THE COEFFICIENTS OF THE POLYNOMINAL IS CALCULATED BY GAUSSIAN
C     ELIIMINATION.  THIS PROGRAM IS FURNISHED BY MR. BING KWAN
C     ELIM(AB,3,4.3)
C
      IF ((ABS(D2)).LT.TOLER) GO TO 555
      AB(1.2)=((W1*R)+(TOLER))/C
      AB(2.2)=(W1*R/C)
      AB(3,2)=((W1*R)-(TOLER))/C
      GO TO 777
555   AB(1.2)=-((TOLER)+(W1*R))/C
      AB(2.2)=-W1*R/C
      AB(3,2)=((TOLER)-W1*R)/C
777   AB(1.4)=FUNC1(R,W1,(AB(1,2))/R)
      AB(2.4)=FUNC2(R,(AB(2.2))/R)
      AB(3,4)=FUNC1(R,W1,(AB(3,2))/R)
      DO 800 J=1,3
             AB(J,3)=1.0
```

```
            AB(J,1)=((AB(J,2))**2.)
800   CONTINUE
      CALL ELIM(AB,3,4,3)
      CONST1=(FUNC3(AB(1,4),AB(2,4),AB(3,4),W2*R))
      GO TO 1000
888   CONST1=FUNC2(R,W2)
      GO TO 1000
999   CONST1=FUNC1(R,W1,W2)
1000  RETURN
      END
```

```
C
C     FUNCTION:       FUNC1
C
C     THIS IS THE RECONSTRUCTION FUNCTION WHEN THE DENOMINATOR IS
C     NOT ZERO.
C
      FUNCTION FUNC1(R,W1,W2)
      C=SQRT(3.)
      D1=(W1-C*W2)*R
      D2=(W1+C*W2)*R
      X1=(COS(R*W1/C))*(COS(R*W2))
      X2=-COS(2.*R*W1/C)
      X3=-(W2*R)*(SINC(R*W1/C))*(SIN(R*W2))
      FUNC1=2.*(X1+X2+X3)/(D1*D2)
      RETURN
      END
```

```
C
C     FUNCTION:       FUNC2
C
C     THIS IS THE RECONSTRUCTION  FUNCTION WHEN W1**2 - 3*W2**2 =0
C
      FUNCTION FUNC2(R,W2)
      C1=(2./3.)*SINC(2.*R*W2)
      C2=(1./3.)*(SINC(R*W2))**2.
      FUNC2=C1+C2
      RETURN
      END
```

```
C
C     FUNCTION:      FUNC3
C
C     THIS FUNCTION WILL PRODUCE THE LINEAR INTERPOLATION PORTION
C     OF THE RECONSTRUCTION FUNCTION
C     WHEN THE UNDERFLOW PROBLEM OCCURS
C
      FUNCTION FUNC3(A,B,C,W2)
      X1=A*(W2**2)
      X2=B*W2
      FUNC3=X1+X2+C
      RETURN
      END
```

```
C
C     FUNCTION:       CONST2
C
C     THIS FUNCTION IS TO RECONSTRUCT RECTANGLUAR SAMPLING
C
C     CALCULATING:
C     (SIN(.5*(V1.W)))*(SIN(.5*(V2.W))/(.25*(V1.W)*(V2.W))
C
C     THIS FUNCTION REQUIRES THE SUPPORT OF SINC
C
      FUNCTION CONST2(V1.V2.W)
      COMPLEX V1.V2.W
      X1=REAL(V1)*REAL(W)+AIMAG(V1)*AIMAG(W)
      X2=REAL(V2)*REAL(W)+AIMAG(V2)*AIMAG(W)
      CONST2=SINC(X1/2.)*SINC(X2/2.)
      RETURN
      END
```

```
C
C     SUBROUTINE:      RPLOT
C
C     THIS SUBROUTINE WILL DO RECTANGULAR PLOT
C     FOR REAL AND IMAGINERY PLOT OR MAGNITUDE AND PHASE PLOT.
C
C     DATA     :COMPLEX ARRAY TO BE PLOTTED(SIZE=360)
C     LAST     :LAST ELEMENT IN THE ARRAY
C     FMIN     :SMALLEST VALUE ON THE ABSCISSOR
C     FINCR    :INCREMENT ON THE ABSCISSOR
C
      SUBROUTINE RPLOT(DATA,LAST,FMIN.FINCR)
      CHARACTER COM
      COMPLEX DATA(360)
      DIMENSION YAXIS1(201),YAXIS2(201),XAXIS(201)
5     WRITE(6,*) 'DO YOU WANT REAL AND IMAGINERY PLOT?Y/N'
      ACCEPT 6, COM
6     FORMAT(A1)
      IF (COM.EQ.'N') GO TO 15
      IF (COM.NE.'Y') GO TO 5
      DO  10 I=1,LAST
              YAXIS1(I)=REAL(DATA(I))
              YAXIS2(I)=AIMAG(DATA(I))
              XAXIS(I)=(I-1)*FINCR + FMIN
10    CONTINUE
      GO TO 25
15    DO 20 I=1,LAST
              XAXIS(I)=(I-1)*FINCR +FMIN
              YAXIS1(I)=CABS(DATA(I))
              IF(REAL(DATA(I)).EQ.0.) GO TO 18
              YAXIS2(I)=ATAN2(AIMAG(DATA(I)),REAL(DATA(I)))
              GO TO 20
18            YAXIS2(I)=0.
20    CONTINUE
25    CALL PLTPKG(XAXIS.YAXIS1,201.201.1.0.1)
      CALL PLTPKG(XAXIS.YAXIS2.201.201.1.0.1)
      RETURN
      END
```

```
C
C     SUBROUTINE      :ELIM
C
C     THIS SUBROUTINE IS COURTESY OF MR. BING KWAN[16]
C
      SUBROUTINE ELIM(AB,N,NP,NDIM)
      DIMENSION AB(NDIM,NP)
C   THIS SUBROUTINE SOLVES A SET OF LINEAR EQUATIONS.
C   THE GAUSS ELIMINATION METHOD IS USED, WITH PARTIAL PIVOTING.
C   MULTIPLE RIGHT HAND SIDES ARE PERMITTED, THEY SHOULD BE SUPPLIED
C   AS COLUMNS THAT AUGMENT THE COEFFICIENT MATRIX.
C   PARAMETERS ARE:
C     AB  COEFFICIENT MATRIX AUGMENTED WITH R.H.S. VECTORS.
C     N   NO. OF EQUATIONS
C     NP  TOTAL NO. OF COLUMNS IN AB
C     NDIM  NO. OF ROWS IN AB
C
C   BEGINS THE REDUCTION
      NM1=N-1
      DO 35 I=1,NM1
C   FIND THE ROW NUMBER OF THE PIVOT ROW. WE WILL THEN
C   INTERCHANGE ROWS TO THE PIVOT ELEMENT ON THE DIAGONAL.
      IPVT=I
      IP1=I+1
      DO 10 J=IP1,N
        IF(ABS(AB(IPVT,I)).LT.ABS(AB(J,I))) IPVT=J
10    CONTINUE
C   CHECK TO BE SURE THE PIVOT ELEMENT IS NOT TOO SMALL, IF SO
C   PRINT A MESSAGE AND RETURN.
      IF(ABS(AB(IPVT,I)).LT.1.E-5) GO TO 99
C   NOW INTERCHANGE , EXCEPT IF THE PIVOT ELEMENT IS ALREADY ON
C   THE DIAGONAL, DO NOT NEED TO.
      IF(IPVT.EQ.I) GO TO 25
      DO 20 JCOL=I,NP
        SAVE=AB(I,JCOL)
        AB(I,JCOL)=AB(IPVT,JCOL)
        AB(IPVT,JCOL)=SAVE
20    CONTINUE
C   NOW REDUCE ALL ELEMENTS BELOW THE DIAGONAL IN THE I-TH ROW.
C   CHECK FIRST TO SEE IF ZERO ALREADY PRESENT. IF SO
C   CAN SKIP REDUCTION FOR THAT ROW.
25    DO 32 JROW=IP1,N
        IF(AB(JROW,I).EQ.0.0) GO TO 32
        RATIO=AB(JROW,I)/AB(I,I)
        DO 30 KCOL=IP1,NP
          AB(JROW,KCOL)=AB(JROW,KCOL)-RATIO*AB(I,KCOL)
30      CONTINUE
32    CONTINUE
35    CONTINUE
C   WE STILL NEED TO CHECK A(N,N) FOR SIZE.
      IF(ABS(AB(N,N)).LT.1.E-5) GO TO 99
```

```
C   NOW WE BACK SUBSTITUTE.
      NP1=N+1
      DO 50 KCOL=NP1,NP
        AB(N,KCOL)=AB(N,KCOL)/AB(N,N)
        DO 45 J=2,N
          NVBL=NP1-J
          L=NVBL+1
          VALUE=AB(NVBL,KCOL)
          DO 40 K=L,N
          VALUE=VALUE-AB(NVBL,K)*AB(K,KCOL)
40        CONTINUE
        AB(NVBL,KCOL)=VALUE/AB(NVBL,NVBL)
45      CONTINUE
50    CONTINUE
      RETURN
C   MESSAGE FOR A NEAR SINGULAR MATRIX.
99    WRITE(66,100)
100   FORMAT(/,'SOLUTION NOT POSSIBE. A NEAR 0 PIVOT ENCOUNTERED.')
      RETURN
      END
```

```
C
C     SUBROUTINE:     WFILE
C
C     THIS SUBROUTINE WILL WRITE A COMPLEX 'DATA' ARRAY OF SIZE
C     'MSIZE' INTO A FILE 'FNAME1'. THE NUMBER OF NON ZERO
C     ELEMENT IS SPECIFIED  BY 'NUM'.
C
      SUBROUTINE WFILE(DATA,MSIZE,NUM)
      COMPLEX DATA(MSIZE)
      CHARACTER*10 FNAME1
      CHARACTER WRI
10    WRITE (6,*) 'DO YOU WANT TO WRITE INTO A FILE?Y/N'
      ACCEPT 9010, WRI
      IF (WRI.EQ.'N') GO TO 9999
      IF (WRI.NE.'Y') GO TO 10
      WRITE(6,*) 'STORAGE FILE NAME:'
      ACCEPT 9020,FNAME1
      OPEN(UNIT=8,NAME=FNAME1,TYPE='NEW')
      DO  30 I=1,NUM
            WRITE(8,*) DATA(I)
30    CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
9010  FORMAT(A1)
9020  FORMAT(A10)
9999  RETURN
      END
```

```
C
C     PROGRAM :       INTEGFFT
C
C     THIS PROGRAM WILL READ FROM A COMPLEX DATA FILE:FNAME1(MSIZE)
C     WHICH IS PART OF AN INTERGRAND FOR INTEGRATION.
C     THE DATA ARGUMENT HAS A RANGE OF [0.360).
C     (I.E. MSIZE IS USUALLY A MULTIPLE OF 360 <680)
C     FOR 8192>MSIZE>512 CHANGE M,NP1 AND THE ARRAY SIZE OF: DATA1.DATA2.S
C     THE OTHER PART OF THE INTEGRAND IS SPECIFED IN
C     FUNC1.  THE INTEGRATION IS LIKE A CONVOLUTION.
C     THEREFORE, THE INTEGRATION IS USING FFT APPROACH
C
C     A RECTANGULAR TO POLAR COORDINATE FILE MUST BE SUPPLIED
C     THIS IS FOR THE TIME PLOT WHICH IS  FNAME3.
C     THIS FILE IS GENERATED BY PROGRAM:RECTPOLAR
C     FNAME3 FORMAT:
C             NUMBER OF POINT:(*)
C             RADIUS.ANGLE(RADIAN),X-COORDINATE,Y-COORDINATE(4E15-8)
C
C     THE ONLY PART THAT IS RELATED TO THE MENSA'S ACTUAL INTEGRATION IS
C     SHOWN AT A LATER SET OF COMMENT BEFORE TEMP1 IS CALCULATED
C
C     LINK INTEGFFT,APTORI,EXPAND,PICK,'SSP
C
      COMPLEX DATA1(512),DATA2(512),FACTOR,POLAR(10000),RECT(10000),DUMMY
      DIMENSION S(128)
      CHARACTER*10 FNAME1.FNAME2.FNAME3
      CHARACTER ONE,ODD
      M=9
      NP1=512
      NPL=129
      NPOLAR=10000
      PI=3.14159265
C
C     XNOR IS USED TO NORMALIZE THE AMP OF THE OUTPUT
C
      XNOR=1E-9
C
C     THIS TIME FACTOR IS EACH UNIT OF THE TIME PLOT
C
      TIME=1/(0.49212598E9*31.)
      WRITE(6,*) 'FILE NAME OF MEASURED DATA:'
      ACCEPT 9000,FNAME1
      WRITE(6,*) 'FREQUENCY USED:(HZ)'
      ACCEPT*,FREQ
      WRITE(6,*) 'NUMBER OF POINTS IN THE FILE:'
      ACCEPT*,MSIZE
C
C     IF THE NUMBER OF POINTS IN THE FILE IS LESS THAN NPL,
C     THEN THE DATA MAY BE ONE QUADRANT DATA ONLY
C
```

```
      IF(MSIZE.GT.NP1) GO TO 998
      IF (MSIZE.GT.NPL) GO TO 110
120   WRITE(6,*) 'IS THIS ONE QUADRANT DATA?Y/N'
      ACCEPT 9020,ONE
      IF (ONE.EQ.'N') GO TO 110
      IF (ONE.NE.'Y') GO TO 120
130   WRITE(6,*) 'COMPLEX CONJUGATE IN 2ND AND 3RD QUADRANTS?Y/N'
      ACCEPT 9020,ODD
      IF ((ODD.NE.'Y').AND.(ODD.NE.'N')) GO TO 130
C
C     TO GUARD AGAINST POSSIBLITY OF MEASUREMENT IN OTHER FORMAT
C
110   WRITE(6,*) 'FORMAT OF DATA:      1)REAL AND IMAGINARY'
      WRITE(6,*) '                     2)AMP(LINEAR) AND PHASE(RADIAN)'
      WRITE(6,*) '                     3)AMP(LINEAR) AND PHASE(DEGREE)'
      WRITE(6,*) '                     4)AMP(DB)     AND PHASE(RADIAN)'
      WRITE(6,*) '                     5)AMP(DB)     AND PHASE(DEGREE)'
      ACCEPT *,NFORM
      IF((NFORM.GT.5).OR.(NFORM.LT.1)) GO TO 110
      WRITE(6,*) 'STORAGE FILE NAME:'
      ACCEPT 9000,FNAME2
      WRITE(6,*) 'NAME OF THE RECTANGULAR TO POLAR FILE:'
      ACCEPT 9000,FNAME3
C
C     INITIALIZATION
C
      DO 10 I=1,NP1
            DATA1(I)=(0.,0.)
10    CONTINUE
C
C     READ IN MEASUREMENT DATA
C
      OPEN(UNIT=8,NAME=FNAME1,TYPE='OLD')
      DO 60 I=1,MSIZE
            READ(8,9030) DUMMY
            CALL APTORI(DUMMY,NFORM)
            DATA1(I)=DUMMY
60    CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
C
C     EXPAND TO FILL THE HALF OF THE SPAN OF THE FFT REPETITIVE UNIT
C
      IF (MSIZE.LE.NPL)CALL EXPAND(DATA1,DATA2,NP1,MSIZE,NPL)
      IF (MSIZE.GT.NPL)CALL EXPAND(DATA1,DATA2,NP1,MSIZE,NP1)
      IF (MSIZE.GT.NPL) GO TO 170
C
C     CASE OF FULL PLANE DATA INPUT
C
      IF (ONE.EQ.'N') CALL EXPAND(DATA1,DATA2,NP1,NPL,NP1)
      IF (ONE.EQ.'N') GO TO 170
C
```

```
C     FILL IN THE OTHER HALF OF THE SPAN BY COMPLEX CONJUGATE OR
C     THE SAME
C
      MSIZE=NPL
      DO 160 I=1,MSIZE-2
              IF (ODD.EQ.'N')DATA1(I+MSIZE)=DATA1(MSIZE-I)
              IF(ODD.EQ.'Y') DATA1(I+MSIZE)=CONJG(DATA1(MSIZE-I))
160   CONTINUE
      MSIZE=2*MSIZE-2
      DO 165 I=1,MSIZE
              IF(ODD.EQ.'N') DATA1(I+MSIZE)=DATA1(I)
              IF (ODD.EQ.'Y') DATA1(I+MSIZE)=CONJG(DATA1(I))
165   CONTINUE
      MSIZE=MSIZE*2
      IF(MSIZE.GT.NP1) GO TO 998
170   MSIZE=NP1
C
C     DELTA   :SIZE OF EACH ANGLE INCREMENT
C
      DELTA=2.*PI/MSIZE
      OPEN(UNIT=8,NAME=FNAME3,TYPE='OLD')
      READ(8,*) NPOINT
      IF (NPOINT.GT.NPOLAR) GO TO 998
C
C     READ THE RECTANGULAR TO POLAR COORDINATE CONVERSION FILE
C
      DO 100 I=1,NPOINT
              READ(8,9005) POLAR(I),RECT(I)
100   CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
      CALL FORT(DATA1,M,S,1,IFERR)
C
C     THE VALUE OF AMP1 =-1 IS ONLY A DUMMY TO START THE ROUTINE
C
      AMP1=-1.
      DO 200 J=1,NPOINT
              WRITE(6,*) 'WORKING ON:',J,' LAST ONE:',NPOINT
              IF (REAL(POLAR(J)).EQ.AMP1) GO TO 50
              AMP1=POLAR(J)
              DO 17 I=1,NP1
                      DATA2(I)=(0.,0.)
17            CONTINUE
              DO 20 I=1,MSIZE
                      THETA=(I-1)*DELTA
                      PHI=2.*PI*(REAL(POLAR(J)))*TIME*FREQ*COS(THETA)
C
C             FUNC1: EXP(JWT)
C             THEREFORE, XREAL=COS(PHI)
C                            YIMAG=SIN(PHI)
C
                      XREAL=COS(PHI)
```

```
                YIMAG=SIN(PHI)
                DATA2(I)=CMPLX(XREAL,YIMAG)
20          CONTINUE
C
C     GO TO FREQUENCY DOMAIN
C
            CALL FORT(DATA2.M,S.2.IFERR)
C
C     CONVOLUTION IN TIME <=> MULTIPLICATION IN FREQUENCY
C
            DO 25 I=1.NP1
                DATA2(I)=DATA1(I)*DATA2(I)
25          CONTINUE
C
C     GO BACK TO TIME DOAMIN
C
            CALL FORT(DATA2.M,S.-2.IFERR)
C
C     NOW POLAR(J) IS THE INTEGRAL VALUE OF 2D FOURIER TRANSFORM
C     WITH IMPULSIVE RADIUS VALUE (MENSA'S INTEGRAL)
C
50          ANGLE=AIMAG(POLAR(J))
            POLAR(J)=PICK(DATA2.NP1.0.,DELTA,ANGLE)
            POLAR(J)=(POLAR(J)/FLOAT(MSIZE))*FREQ*XNOR
200   CONTINUE
C
C     WRITE OUT THE FILE
C     FILE FORMAT:
C     NPOINT:NUMBER OF POINTS(I10)
C     REAL,IMAGINARY,X-COORDINATE,Y-COORDINATE (4E15.8)
C
      OPEN(UNIT=9,NAME=FNAME2.TYPE='NEW')
      WRITE(9,9010) NPOINT
      DO 300 I=1.NPOINT
            WRITE(9.9005) POLAR(I),RECT(I)
300   CONTINUE
      CLOSE(UNIT=9,DISP='SAVE')
      GO TO 999
998   WRITE(6,*) 'ERROR: SIZE OF FILE TOO LARGE!'
C
C     FORMAT STATEMENTS
C
9000  FORMAT(A10)
9005  FORMAT(4E15.8)
9010  FORMAT(I10)
9020  FORMAT(A1)
9030  FORMAT(2E15.8)
C
999   STOP
      END
```

```
C
C       SUBROUTINE:     EXPAND
C
C       THIS SUBROUTINE WILL EXPAND AN ARRAY DATA(MSIZE) CONTAINING 'NFILL'
C       DATA ELEMENTS INTO 'NEXP'  DATA ELEMENT USING LINEAR INTERPOLATION
C       2<NFILL<NEXP
C
C       WORK: A DUMMY ARRAY FOR TEMPORARY STORAGE
C
        SUBROUTINE EXPAND(DATA,WORK,MSIZE,NFILL,NEXP)
        COMPLEX DATA(MSIZE),WORK(MSIZE),DIFF
        ERROR=1E-6
        DELTA1=1./FLOAT(NFILL-1)
        DELTA2=1./FLOAT(NEXP-1)
        DO 10 I=1,NFILL
               WORK(I)=DATA(I)
10      CONTINUE
        DATA(1)=WORK(1)
        DATA(NEXP)=WORK(NFILL)
        DO 20 I=1,NEXP-2
               X=(I)*DELTA2
               N=INT(X/DELTA1)+1
               REMAIN=AMOD(X,DELTA1)
               RATIO=(X-DELTA1*FLOAT(N-1))/DELTA1
               DIFF=(WORK(N+1)-WORK(N))*RATIO
               DATA(I+1)=WORK(N)+DIFF
20      CONTINUE
        RETURN
        END
```

```
C
C     COMPLEX FUNCTION:       PICK
C
C     THIS FUNCTION WILL RETURN A LINEAR INTERPOLATED COMPLEX
C     VALUE BACK.
C     DATA: ARRAY
C     MSIZE: SIZE OF THE ARRAY
C     FIRST: FIRST/DELTA IS THE FIRST INDEX
C     DELTA: INCREMENT OF EACH STEP IN THE ARRAY
C     VALUE: VALUE/DELTA IS THE LOCATION IN THE ARRAY
C     N       :EXACT LOCATION OR LOCATION -1
C     VARY    :AMOUNT OF ANGLE DIFFERENCE
C     FACTOR  :ADJUSTMENT TO DATA2(N)
C
      COMPLEX FUNCTION PICK(DATA,MSIZE,FIRST,DELTA,VALUE)
      COMPLEX DATA(MSIZE),FACTOR
      IF ((VALUE.LT.FIRST).OR.(VALUE.GT.(MSIZE*DELTA +FIRST))) GO TO 998
      N=INT((VALUE-FIRST)/DELTA) +1
      VARY=(VALUE)-((N-1)*DELTA+FIRST)
      FACTOR=((DATA(N+1)-DATA(N))/DELTA)*VARY
      PICK=DATA(N) +FACTOR
      GO TO 999
998   WRITE(6,*) 'ERROR:INTERPOLATED POINT IS OUTSIDE THE RANGE!'
999   RETURN
      END
```

```
C
C      SUBROUTINE:     APTORI
C
C      THIS SUBROUTINE WILL CONVERT AMPLTIUDE AND PHASE (VALUE) INTO
C      REAL AND IMAGINARY (VALUE).
C       BY SPECIFYING N
C              N=1:VALUE IS ALREADY IN REAL AND IMAGINARY FORM
C              N=2:AMPLITUDE(LINEAR),PHASE(RADIAN)
C              N=3:AMPLITUDE(LINEAR),PHASE(DEGREE)
C              N=4:AMPLITUDE(DB)     ,PHASE(RADIAN)
C              N=5:AMPLITUDE(DB)     ,PHASE(DEGREE)
C
       SUBROUTINE APTORI(VALUE,N)
       COMPLEX VALUE
       IF (N.EQ.1) GO TO 999
       PI=3.14159265
       AMP=REAL(VALUE)
       PHASE=AIMAG(VALUE)
       IF ((N.EQ.4).OR.(N.EQ.5)) AMP=10.0**(AMP/20.)
       IF ((N.EQ.3).OR.(N.EQ.5)) PHASE=PHASE*PI/180.
       VALUE=CMPLX(AMP*COS(PHASE),AMP*SIN(PHASE))
999    RETURN
       END
```

```
C
C     PROGRAM :       RECTPOLAR
C
C     THIS PROGRAM WILL CHANGE RECTANGLAR COORDINATES TO
C     POLAR COORDINATES.  THE FILE 'FNAME' OUTPUTS IN THE
C     FOLLOWING FORMAT:
C            RADIUS,PHI(RADIAN),X,Y (4E15.8)
C     THE FILE WILL ALSO BE ORGANIZED FROM SMALLEST RADIUS
C     TO THE LARGEST RADIUS.
C     NOTE THE MAXIMUM X COORDINATE VALUE: 'MAX' WILL GIVE
C     (MAX*2)**2 +2*2*MAX +1 DATA POINTS
C     THEREFORE A GRID OF 256X256 CAN ACCOMMODATE MAX=127
C
C     LINK RECTPOLAR,GEN.SEARCH1.INSERT1.PUSH.NORMAL
C
      COMPLEX DATA(65536,2)
      CHARACTER*10 FNAME
      WRITE(6,*) 'FILE NAME FOR STORAGE OF RESULTS:'
      ACCEPT 5,FNAME
5     FORMAT(A10)
      WRITE(6,*) 'MAXIMUM X COORDINATE VALUES:'
      ACCEPT *,MAX
      MSIZE=65536
      PI=3.14159265
      M=0
      M=M+1
      DATA(M,1)=CMPLX(0.,0.)
      DATA(M,2)=CMPLX(0.,0.)
C
C     THIS DO LOOP GENERATES ALL THE AXIS POINTS
C
      DO 10 J=1,MAX
             X1=FLOAT(J)
             M=M+1
             DATA(M,1)=CMPLX(X1,0.)
             DATA(M,2)=CMPLX(X1,0.)
             M=M+1
             DATA(M,1)=CMPLX(X1,PI/2)
             DATA(M,2)=CMPLX(0.,X1)
             M=M+1
             DATA(M,1)=CMPLX(X1,PI)
             DATA(M,2)=CMPLX(-X1,0.)
             M=M+1
             DATA(M,1)=CMPLX(X1,3.*PI/2.)
             DATA(M,2)=CMPLX(0.,-X1)
10    CONTINUE
      WRITE(6,*) 'CALCULATING FIRST QUADRANT POINTS!'
      CALL GEN(MAX,DATA,MSIZE,M,1,1)
      WRITE(6,*) 'CALCULATING SECOND QUADRANT POINTS!'
      CALL GEN(MAX,DATA,MSIZE,M,-1,1)
      WRITE(6,*) 'CALCULATING THIRD QUADRANT POINTS!'
```

```
      CALL GEN(MAX,DATA,MSIZE,M,-1,-1)
      WRITE(6,*) 'CALCULATING FOURTH QUADRANT POINTS!'
      CALL GEN(MAX,DATA,MSIZE,M,1,-1)
      OPEN (UNIT=8,NAME=FNAME,TYPE='NEW')
      WRITE(8,*) M
      DO 20 I=1,M
            WRITE(8,1000) DATA(I,1),DATA(I,2)
20    CONTINUE
1000  FORMAT(4E15.8)
      STOP
      END
```

```
C
C     SUBROUTINE:     GEN
C
C     THIS SUBROUTINE WILL GENERATE OFF AXIS POINTS
C     FOR V1=(1,0),V2=(0.1)
C     THIS IS A MODIFIED VERSION OF GEN1 IN PTGRID
C
C     MAX      :MAXIMUM INDEX
C     DFILE    :COMPLEX ARRAY FOR STOAGE
C     MSIZE    :DIMENSION OF DFILE
C     M        :NUMBER OF ELEMENT
C     N1SIGN   :SIGN OF N1  (N1*V1)
C     N2SIGN   :SIGN OF N2  (N2*V2)
C
C     THIS REQUIRES THE SUPPORT OF SEARCH1.INSERT1,NORMAL
C
      SUBROUTINE GEN(MAX.DFILE,MSIZE,M,N1SIGN.N2SIGN)
      EXTERNAL SEARCH1
      COMPLEX DFILE(MSIZE,2),VECTOR,V1.V2
      REAL MAG
      V1=CMPLX(1.,0.)
      V2=CMPLX(0.,1.)
      DO 5 I=1.MAX
             WRITE(6,*) I
             DO 10 J=1.MAX
                    VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
                    MAG=CABS(VECTOR)
                    PHASE=ATAN2(AIMAG(VECTOR),REAL(VECTOR))
                    CALL NORMAL(PHASE)
                    VECTOR=CMPLX(MAG,PHASE)
                    LOC=SEARCH1(VECTOR,DFILE,MSIZE,M)
                    RJ=FLOAT(J)
                    RI=FLOAT(I)
                    IF ((N1SIGN.LT.0).AND.(N2SIGN.LT.0))
     1                       CALL INSERT1(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
                    IF ((N1SIGN.LT.0).AND.(N2SIGN.GT.0))
     2                       CALL INSERT1(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
                    IF ((N1SIGN.GT.0).AND.(N2SIGN.LT.0))
     3                       CALL INSERT1(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
                    IF ((N1SIGN.GT.0).AND.(N2SIGN.GT.0))
     4                       CALL INSERT1(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)
10           CONTINUE
5     CONTINUE
      RETURN
      END
```

```
C
C     FUNCTION:        SEARCH1
C
C     THIS FUNCTION WILL SEARCH THE LOCATION WHERE THE ELEMENT
C     FITS INTO A DATA ARRAY.  THE ARRAY ASSUMES AT LEAST TWO
C     MEMBERS.  THE ELEMENT MAY HAVE 3 POSSIBILITIES:-
C             1) THE REAL PART IS GREATER THAN THE RETURNED LOCATION
C             2) THE REAL PART IS THE SAME
C                     BUT THE IMAGINARY PART >RETURNED LOCATION VALUES
C             3) THE REAL AND IMAGINARY PART ARE THE SAME
C                     AS THE RETURNED LOCATION VALUES
C
C     ELEMENT :COMPLEX ELEMENT TO BE INSERTED
C     DFILE   :COMPLEX ARRAY TO BE SEARCHED
C     MSIZE   :SIZE OF DFILE
C     LAST    :NUMBER OF NON-ZERO ELEMENT IN DFILE
C
      FUNCTION SEARCH1(ELEMENT,DFILE,MSIZE,LAST)
      COMPLEX DFILE(MSIZE,2),ELEMENT
      LOC2=LAST
      LOC1=1
      IF (REAL(ELEMENT).LT.REAL(DFILE(LOC1,1))) GO TO 147
      IF (REAL(ELEMENT).GT.REAL(DFILE(LOC2,1))) GO TO 140
      IF ((REAL(ELEMENT).EQ.REAL(DFILE(LOC1,1))).AND.
     1                (AIMAG(ELEMENT).LT.AIMAG(DFILE(LOC1,1)))) GO TO 147
      IF ((REAL(ELEMENT).EQ.REAL(DFILE(LOC2,1))).AND.
     1                (AIMAG(ELEMENT).GT.AIMAG(DFILE(LOC2,1)))) GO TO 140
120   LOCM = (LOC1 +LOC2)/2
      IF(LOCM.EQ.LOC1) GO TO 150
      IF (REAL(ELEMENT).LT.REAL(DFILE(LOCM,1))) GO TO 125
      IF (REAL(ELEMENT).GT.REAL(DFILE(LOCM,1))) GO TO 130
      IF (AIMAG(ELEMENT).LT.AIMAG(DFILE(LOCM,1))) GO TO 125
      IF (AIMAG(ELEMENT).GT.AIMAG(DFILE(LOCM,1))) GO TO 130
      GO TO 145
125   LOC2 =LOCM
      GO TO 120
130   LOC1=LOCM
      GO TO 120
140   SEARCH1 = LOC2
      GO TO 155
145   SEARCH1 = LOCM
      GO TO 155
147   SEARCH1 = LOC1-1
      GO TO 155
150   SEARCH1 = LOC1
155   RETURN
      END
```

```
C
C     SUBROUTINE:     INSERT1
C
C     THIS SUBROUTINE WILL INSERT THE ELEMENT INTO THE DFILE
C     IF THE ELEMENT HAS A DIFFERENT REAL AND/OR IMAGINARY PART.
C     THE REAL PART HAS A SENSITIVITY SPECIFIED BY THE ERROR1.
C     THE IMAGINARY PART HAS A SENSITIVITY SPECIFIED BY ERROR2.
C     THE PRIORITY IS REAL PART FIRST THEN IMAGINARY PART.
C
C     NSTART  :LOCATION OF THE ELEMENT
C     LAST    :NUMBER OF NON-ZERO ELEMENTS IN DFILE
C     DFILE   :ARRAY FILE TO BE INSERTED
C     MSIZE   :SIZE OF DFILE
C     ELEMENT :COMPLEX ELEMENT TO BE INSERTED
C     V1INDEX :NUMBER OF V1 USED
C     V2INDEX :NUMBER OF V2 USED
C
      SUBROUTINE INSERT1(NSTART,LAST,DFILE,MSIZE,ELEMENT,V1INDEX,V2INDEX)
      COMPLEX DFILE(MSIZE,2),ELEMENT
      PI=3.14159265
      ERROR1= 1.E-6
      ERROR2=0.01*PI/180.
      IF (NSTART.NE.LAST) GO TO 200
C
C     CASE OF NSTART=LAST
C
C     IF(ABS(REAL(ELEMENT)-REAL(DFILE(LAST,1))).GT.ERROR1) GO TO 190
      IF(REAL(ELEMENT).GT.REAL(DFILE(LAST,1))) GO TO 190
      IF(ABS(AIMAG(ELEMENT)-AIMAG(DFILE(LAST,1))).LT.ERROR2) GO TO 295
190   LAST = LAST +1
      DFILE(LAST,1)=ELEMENT
      DFILE(LAST,2)=CMPLX(V1INDEX,V2INDEX)
      GO TO 295
200   IF (NSTART.EQ.0) GO TO 202
C     IF (ABS(REAL(ELEMENT)-REAL(DFILE(NSTART,1))).GT.ERROR1) GO TO 202
      IF (REAL(ELEMENT).GT.REAL(DFILE(NSTART,1))) GO TO 202
      IF (ABS(AIMAG(ELEMENT)-AIMAG(DFILE(NSTART,1))).LT.ERROR2) GO TO 295
202   CALL PUSH(NSTART+1,LAST,DFILE,MSIZE)
      DFILE(NSTART+1,1)=ELEMENT
      DFILE(NSTART+1,2)=CMPLX(V1INDEX,V2INDEX)
295   RETURN
      END
```

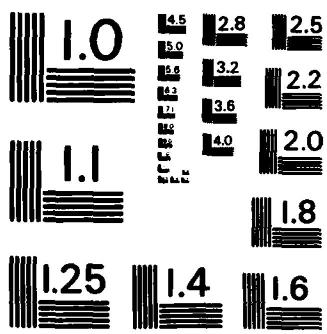
```
C
C     PROGRAM:      SUM3D
C
C     THIS PROGRAM CAN DO THE FOLLOWING:
C             1) READ A DATA FILE
C             2) SUMMING OF DATA FILES
C             3) PLOT A DATA FILE IN A)3-D PICTURE PLOT
C                                     B)CONTOUR PLOT
C             4) 2-D FOURIER TRANSFORM
C             5) MODIFIES FREQUENCY DATA BY A)1/JW
C                                          B)-1/W**2
C             6) COSINE TAPERING LOW PASS THE FREQUENCY DATA
C             7) HIGH PASS THE FREQUENCY DATA
C             8) WRITE OUT THE FILE
C
C     THE INPUT (OR OUTPUT) FILE FORMAT:
C             NPOINT:NUMBER OF POINTS IN THE FILE(I10)
C             DATA,X-COORDINATE,Y-COORDINATE(4E15.8)
C
C     THE CONTOUR PLOT LINKS WITH NCAR PLOTTING PACKAGES
C     THE LINKING REQUIRED IS:
C     $CASSIGN DRA2:[NCAR2.NCARPLOT.PLOTLIB]  NCAR_PLOT_LIB
C     $ASSIGN DRA2:[NCAR2.NCARLIB]            NCAR_LIB
C     $ASSIGN DRA2:[NCAR2.NCARPLOT]           NCAR_PLOT
C     $ASSIGN DRA2:[NCAR2.NCARPLOT.CHROME]    NCAR_PLOT_CHROME
C     $ASSIGN DRA2:[NCAR2.NCARPLOT.TEST]      NCAR_PLOT_TEST
C     $ASSIGN DRA2:[NCAR2.NCARPLOT.DOC]       NCAR_PLOT_DOC
C     $ASSIGN DRA2:[NCAR2.NCARPLOT.TRANS]     NCAR_PLOT_TRANS
C     $LINK SSUM3D,DPLOT,TRAN2D,WEIGHT,PLOT3D,APTORI,'SSP,'PLOTLIB,-
C     DRA2:[NCAR2.NCARPLOT.PLOTLIB]NCARCONRE/LIB,NCARDASHC/LIB,-
C     NCARGRAPP/LIB,NCARGRAPH/LIB,[NCAR2.NCARLIB]UTILITY/LIB
C
      COMPLEX DATA(31,31),DUMMY,APTORI,CJ
      DIMENSION ARRAY1(31,31),ARRAY2(31.31),ARRAYP(31,31)
      CHARACTER*10 FNAME1.FNAME2
      CHARACTER ADD.OMEGA,SPACE,LP,HP
      N1=5
      MSIZE=2**N1-1
      PI=3.14159265
      CJ=CMPLX(0.1)
      K=MSIZE/2+1
C
C     INITIALIZATION
C
      DO 5 I=1,MSIZE
              DO 6  J=1,MSIZE
                      DATA(I,J)=(0.,0.)
6             CONTINUE
5     CONTINUE
8     WRITE(6,*) 'DATA FILE NAME:'
      ACCEPT 9000,FNAME1
```

```
110   WRITE(6,*) 'WHAT IS THE DATA FORMAT:    1) REAL AND IMAG'
      WRITE(6,*) '                           2) AMP(LINEAR) AND PHASE(RADIAN)'
      WRITE(6,*) '                           3) AMP(LINEAR) AND PHASE(DEGREE)'
      WRITE(6,*) '                           4) AMP(DB)     AND PHASE(RADIAN)'
      WRITE(6,*) '                           5) AMP(DB)     AND PHASE(DEGREE)'
      ACCEPT *,NCOM
      IF ((NCOM.GT.5).OR.(NCOM.LT.1)) GO TO 110
C
C     READ IN THE DATA
C
      OPEN(UNIT=10,NAME=FNAME1,TYPE='OLD')
      READ(10,9010) NPOINT
      DO 10 I=1,NPOINT
            READ(10,9020) DUMMY,X1,Y1
            CALL APTORI(DUMMY,NCOM)
            M=INT(X1) +K
            N=INT(Y1) +K
            DATA(M,N)= DATA(M,N) +DUMMY
10    CONTINUE
      CLOSE(UNIT=10,DISP='SAVE')
C
C     PLOT
C
      CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
C
C     ADD ANOTHER FILE
C
20    WRITE(6,*) ' ADD ANOTHER FILE?  Y/N'
      ACCEPT 9030,    ADD
      IF (ADD.EQ.'Y') GO TO 8
      IF (ADD.NE.'N') GO TO 20
C
C     WRITE INTO A FILE
C
25    WRITE(6,*) ' WRITE YOUR DATA INTO A FILE?  Y/N'
      ACCEPT 9030,KEEP
      IF (KEEP.EQ.'N') GO TO 30
      IF (KEEP.NE.'Y') GO TO 25
      WRITE(6,*) 'FILE NAME:'
      ACCEPT 9000,FNAME2
      OPEN (UNIT=8,NAME=FNAME2,TYPE='NEW')
      WRITE(8,9010) MSIZE**2
      DO  27 I=1,MSIZE
            WRITE(8,9020), (DATA(I,J),FLOAT(I-K),FLOAT(J-K),J=1,MSIZE)
27    CONTINUE
      CLOSE(UNIT=8,DISP='SAVE')
C
C     FOURIER TRANSFORM
C
30    WRITE(6,*) ' GO TO FREQUENCY DOMAIN?  Y/N'
      ACCEPT 9030,OMEGA
```

```
      IF (OMEGA.EQ.'N') GO TO 50
      IF (OMEGA.NE.'Y') GO TO 30
      CALL TRAN2D(DATA,MSIZE,-1)
      CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2.ARRAYP)
      GO TO 25
C
C     INVERSE FOURIER TRANSFORM
C
50    WRITE(6,*) ' GO TO TIME DOMAIN? Y/N'
      ACCEPT 9030,TIME
      IF (TIME.EQ.'N') GO TO 60
      IF (TIME.NE.'Y') GO TO 50
C
C     MODIFY TO PREPARE FOR STEP OR RAMP RESPONSE
C
120   WRITE(6,*) ' WANT TO MODIFY DATA? BY     1)1'
      WRITE(6,*) '                             2)1/JW'
      WRITE(6,*) '                             3)-1/W**2'
      ACCEPT *,MOD
      IF ((MOD.GT.3).OR.(MOD.LT.1)) GO TO 120
      IF (MOD.EQ.1)   GO TO 95
      IF (MOD.EQ.3) GO TO 130
C
C     MODIFY BY 1/JW
C
      DO 125  I=1,MSIZE
              DO 127 J=1,MSIZE
                      W=2.*PI*SORT(FLOAT((I-MSIZE/2-1)**2+(J-MSIZE/2-1)**2))
                      IF(W.NE.0.) DATA(I,J)=DATA(I,J)/(CJ*W)
                      IF (W.EQ.0.) DATA(I,J)=(0.,0.)
127           CONTINUE
125   CONTINUE
      GO TO 145
C
C     MODIFY BY -1/W**2
C
130   DO 135 I=1,MSIZE
              DO 140 J=1.MSIZE
                      W=2.*PI*SORT(FLOAT((I-MSIZE/2-1)**2+(J-MSIZE/2-1)**2))
                      IF (W.NE.0.)DATA(I,J)=-DATA(I,J)/(W**2)
                      IF(W.EQ.0.) DATA(I,J)=(0.,0.)
140           CONTINUE
135   CONTINUE
145   CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2.ARRAYP)
C
C     COSINE TAPERING LOW PASS
C     THIS IS BY SPECIFYING THE NUMBER OF HARMONICS
C
95    WRITE(6,*) 'WANT TO DO COSINE LOW-PASS?Y/N'
      ACCEPT 9030,LP
      IF (LP.EQ.'N') GO TO 100
```

AD-A162 553 SPACE-FREQUENCY SAMPLING CRITERIA FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT(U) OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F Y FOK AUG 85 3/3

UNCLASSIFIED ESL-714190-11 N00014-82-K-0037 F/G 20/14 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      IF (LP.NE.'Y') GO TO 95
      WRITE(6,*) 'CUT-OFF ELEMENT NUMBER:'
      ACCEPT *,NCUT
      DO 200 I=1,MSIZE
              DO 300 J=1,MSIZE
                      DATA(I,J)=DATA(I,J)*WEIGHT(NCUT,I,J,MSIZE)
300           CONTINUE
200   CONTINUE
      CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
C
C     HIGH PASS FILTERING
C     AGAIN BY SPECIFYING THE NUMBER OF HARMONICS
C
100   WRITE(6,*) 'HIGH PASS FILTERING?Y/N'
      ACCEPT 9030,HP
      IF(HP.EQ.'N') GO TO 400
      IF(HP.NE.'Y') GO TO 100
      WRITE(6,*) 'CUT-OFF ELEMENT NUMBER:'
      ACCEPT *,NCUT
      DO 500 I=1,MSIZE
              DO 600 J=1,MSIZE
                RFREQ=SQRT((FLOAT(I-MSIZE/2-1))**2 + (FLOAT(J-MSIZE/2-1))**2)
                IF (RFREQ.LE.FLOAT(NCUT)) DATA(I,J)=(0.,0.)
600   CONTINUE
500   CONTINUE
      CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
C
C     DO FOURIER TRANSFORM NOW
C
400   CALL TRAN2D(DATA,MSIZE,1)
      WRITE(6,*) 'THE DATA ARE IN TIME DOMAIN NOW!'
      CALL DPLOT(DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
      GO TO 25
9000  FORMAT(A10)
9010  FORMAT(I10)
9020  FORMAT(4E15.8)
9030  FORMAT(A1)
60    STOP
      END
```

```
C
C       SUBROUTINE:     DPLOT
C
C       THIS SUBROUTINE WILL PLOT 3-D FOR A COMPLEX ARRAY DATA
C               EITHER IN A 3-D PICTURE OR A CONTOUR PLOT
C       THE CONTOUR PLOT LINKS WITH NCAR PLOTTING PACKAGE
C
C       DATA    :2-D COMPLEX ARRAY TO BE PLOTTED
C       MSIZE   :SIZE OF DATA
C       ARRAY1.ARRAY2.ARRAYP:DUMMY ARRAYS WITH SAME DIMENSIONS AS DATA
C
        SUBROUTINE DPLOT(DATA,MSIZE,ARRAY1.ARRAY2.ARRAYP)
        COMPLEX DATA(MSIZE,MSIZE)
        DIMENSION ARRAY1(MSIZE,MSIZE),ARRAY2(MSIZE,MSIZE),ARRAYP(MSIZE,MSIZE)
        CHARACTER ACT,PLO,COM,COM2,COM3,COM4.COM5.LINE,PICK
10      WRITE(6,*) 'DO YOU WANT TO PLOT?Y/N'
        ACCEPT 9030,COM
        IF (COM.EQ.'N') GO TO 9999
        IF (COM.NE.'Y') GO TO 10
C
C       NORMALIZE THE WHOLE DATA FILE  BY SOME FACTOR
C
500     WRITE(6,*) 'DO YOU WANT TO NORMALIZE WITH A FACTOR?Y/N'
        ACCEPT 9030,COM5
        IF(COM5.EQ.'N') GO TO 510
        IF(COM5.NE.'Y') GO TO 500
        WRITE(6,*) 'FACTOR:'
        ACCEPT *,FACTOR
        GO TO 20
510     FACTOR=1.
20      WRITE(6,*) ' PLOT REAL AND IMAGINARY?  Y/N'
        ACCEPT 9030, ACT
        IF (ACT.EQ.'N') GO TO 50
        IF (ACT.NE.'Y') GO TO 20
        DO 40 I=1,MSIZE
                DO 30 J=1,MSIZE
                        ARRAY1(I,J)=FACTOR*REAL(DATA(I,J))
                        ARRAY2(I,J)=FACTOR*AIMAG(DATA(I,J))
30              CONTINUE
40      CONTINUE
        GO TO 100
C
C       CONVERSION INTO LINEAR AMPLITUDE AND PHASE(RADIAN)
C
50      DO 70 I=1,MSIZE
                DO 60 J=1,MSIZE
                        ARRAY1(I,J)=FACTOR*CABS(DATA(I,J))
                        IF ((REAL(DATA(I,J)).EQ.0.).AND.
                                (AIMAG(DATA(I,J)).EQ.0.)) GO TO 55
                        ARRAY2(I,J)=ATAN2(REAL(DATA(I,J)),AIMAG(DATA(I,J)))
                        GO TO 60
```

```
55                    ARRAY2(I,J)=0.
60             CONTINUE
70    CONTINUE
C
C***************
C
100   WRITE(6,*) '1)3D PICTURE,2)CONTOUR PLOT'
      ACCEPT 9030,PICK
      IF ((PICK.NE.'2').AND.(PICK.NE.'1')) GO TO 100
101   WRITE(6,*) 'PLOT AMPLITUDE/REAL?Y/N'
      ACCEPT 9030,COM4
      IF(COM4.EQ.'N') GO TO 150
      IF(COM4.NE.'Y') GO TO 101
      IF(PICK.EQ.'1') GO TO 102
C
C     CONTOUR PLOT
C
      IF (PICK.EQ.'2')CALL EZCNTR(ARRAY1.MSIZE.MSIZE)
      GO TO 150
C
C     PLOT 3 D PICTURE
C
102   CALL VPLOTS(0.0.0)
120   WRITE(6,*) 'PLOT X-LINE,Y-LINE,OR BOTH?X,Y,B'
      ACCEPT 9030,LINE
      IF ((LINE.NE.'X').AND.(LINE.NE.'Y').AND.(LINE.NE.'B')) GO TO 120
      CALL SET_LINES 3D(LINE)
      CALL SET_ROTATION 3D(0)
      XMAX=ARRAY1(1,1)
      XMIN=ARRAY1(1,1)
      DO 300 I=1.MSIZE
             DO 310  J=1.MSIZE
                    XMAX=AMAX1(ARRAY1(I,J),XMAX)
                    XMIN=AMIN1(ARRAY1(I,J),XMIN)
                    ARRAYP(I,J)=ARRAY1(I,J)
310          CONTINUE
300   CONTINUE
      WRITE(6,*) 'SCALE OF AMPLITUDE/REAL PLOT:'
      WRITE(6,*) '(MAX=',XMAX,'  MIN=',XMIN
      WRITE(6,*) ' RECOMMEND SCALE:'.0.5/(XMAX-XMIN),')'
      ACCEPT *,SCALE
      WRITE(6,*) 'WHAT IS THE VIEW ANGLE W.R.T. X-Y PLANE?'
      WRITE(6,*) '(RECOMMENDING:45)'
      ACCEPT *,VIEW
      CALL PLOT_3D_SURFACE(ARRAYP,MSIZE,MSIZE.VIEW,0.,0.,SCALE)
4001  WRITE(6,*) 'IS THIS THE LAST PLOT?Y/N'
      ACCEPT 9030, PLO
      IF ((PLO.NE.'Y').AND.(PLO.NE.'N')) GO TO 4001
      IF (PLO.EQ.'N')  NSIGN=-1
      IF (PLO.EQ.'Y')  NSIGN=1
      CALL PLOT(0.,0.,NSIGN*999)
```

```
C************
C
C     PLOT IMAGINARY OR PHASE
C
150   WRITE(6,*) 'PLOT THE PHASE/IMAGINARY?Y/N'
      ACCEPT 9030,COM3
      IF (COM3.EQ.'N') GO TO 5000
      IF (COM3.NE.'Y') GO TO 150
      IF (PICK.EQ.'1') GO TO 157
C
C     CONTOUR PLOT
C
      CALL EZCNTR(ARRAY2.MSIZE,MSIZE)
      GO TO 5000
C
C     3 D PICTURE PLOT
C
157   CALL VPLOTS(0.0.0)
220   WRITE(6,*) 'PLOT X-LINE,Y-LINE,OR BOTH?X,Y,B'
      ACCEPT 9030.LINE
      IF ((LINE.NE.'X').AND.(LINE.NE.'Y').AND.(LINE.NE.'B')) GO TO 220
      CALL SET_LINES 3D(LINE)
      CALL SET_ROTATION 3D(0)
C
C     FIND THE MINIMUM AND MAXIMUM IN THE PLOTTING SET
C
      XMAX=ARRAY2(1,1)
      XMIN=ARRAY2(1,1)
      DO 400 I=1,MSIZE
              DO 410  J=1,MSIZE
                      XMAX=AMAX1(ARRAY2(I,J),XMAX)
                      XMIN=AMIN1(ARRAY2(I,J),XMIN)
                      ARRAYP(I,J)=ARRAY2(I,J)
410           CONTINUE
400   CONTINUE
      WRITE(6,*) 'SCALE OF PHASE/IMAGINARY PLOT:'
      WRITE(6,*) '(MAX='.XMAX,'  MIN='.XMIN
      WRITE(6,*) ' RECOMMEND SCALE:',0.5/(XMAX-XMIN),')'
      ACCEPT *,SCALE
      WRITE(6,*) 'WHAT IS THE VIEW ANGLE W.R.T. X-Y PLANE?'
      WRITE(6,*) '(RECOMMENDING:45)'
      ACCEPT *,VIEW
      CALL PLOT_3D SURFACE(ARRAYP,MSIZE,MSIZE,VIEW,0.,0.,SCALE)
4000  WRITE(6,*) 'IS THIS THE LAST PLOT?Y/N'
      ACCEPT 9030, PLO
      IF ((PLO.NE.'Y').AND.(PLO.NE.'N')) GO TO 4000
      IF (PLO.EQ.'N')  NSIGN=-1
      IF (PLO.EQ.'Y')  NSIGN=1
      CALL PLOT(0.,0.,NSIGN*999)
5000  WRITE(6,*) 'PLOT AGAIN?Y/N'
      ACCEPT 9030, COM2
```

```
      IF (COM2.EQ.'Y') GO TO 100
      IF (COM2.NE.'N') GO TO 5000
9030  FORMAT(A1)
9999  RETURN
      END
```

```
C
C     SUBROUTINE:     TRAN2D
C
C     THIS SUBROUNTINE WILL FOURIER TRANSFORM ON A COMPLEX ARRAY IN 2D PLANE.
C     THE PLANE IS ASSUMED TO HAVE THE ORIGIN AT THE CENTRE OF THE PLOT.
C     THE PLANE HAS EQUAL NUMBER OF UNITS ON EACH SIDE OF THE THE AXES.
C     I.E. THE SIZE OF THE ARRAY (MSIZE) IS AN  ODD NUMBER.
C     WORK1 IS THE WORKING ARRAY WITH SIZE 2**N TO JUST FIT THE DATA ARRAY.
C
C     DATA    :COMPLEX 2-D ARRAY TO BE TRANSFORMED, AND RESULT STORED
C     MSIZE   :SIZE OF DATA
C     IFSET   :+1,GO TO TIME
C             :-1,GO TO FREQUENCY
C
      SUBROUTINE TRAN2D(DATA,MSIZE,IFSET)
      COMPLEX WORK1(32.32),DATA(MSIZE,MSIZE)
      INTEGER INV(8),MM(3)
      DIMENSION S(8)
      N=(MSIZE+1)/2
      MSIZE1=32
      MM(1)=5
      MM(2)=5
      MM(3)=0
C
C     PLACE THE CENTRED PICTURE INTO THE FOUR CORNERS
C
      DO 10 I=1,MSIZE1
              DO 5 J=1,MSIZE1
                      WORK1(I,J)=(0.,0.)
5             CONTINUE
10    CONTINUE
      DO 20 I=1,N
              DO 15 J=1,N
                 WORK1(I,J)=DATA(N+I-1,N+J-1)
15            CONTINUE
20    CONTINUE
      DO 30   I=1,N-1
              DO 25   J=1,N-1
                 WORK1(MSIZE1-N+I+1,MSIZE1-N+J+1)=DATA(I,J)
25            CONTINUE
30    CONTINUE
      DO 40   I=1,N
              DO 35 J=1,N-1
                      WORK1(MSIZE1-N+J+1,I)=DATA(J,I+N-1)
                      WORK1(I,J+MSIZE1-N+1)=DATA(I+N-1,J)
35            CONTINUE
40    CONTINUE
C
C     FOURIER TRANSFORM
C
      CALL HARM(WORK1,MM,INV,S,IFSET,IFERR)
```

```
C
C     PUT THE FOUR CORNERS BACK TO THE CENTRE
C
      DO 120 I=1,N
             DO 115 J=1,N
                    DATA(N+I-1,N+J-1)=WORK1(I,J)
115          CONTINUE
120   CONTINUE
      DO 130  I=1,N-1
             DO 125  J=1,N-1
                    DATA(I,J)=WORK1(MSIZE1-N+I+1,MSIZE1-N+J+1)
125          CONTINUE
130   CONTINUE
      DO 140  I=1,N
             DO 135 J=1,N-1
                    DATA(J,I+N-1)=WORK1(MSIZE1-N+J+1,I)
                    DATA(I+N-1,J)=WORK1(I,J+MSIZE1-N+1)
135          CONTINUE
140   CONTINUE
      RETURN
      END
```

```
C
C     FUNCTION:       WEIGHT
C
C     THIS FUNCTION WILL CALCULATED THE COSINE LOW PASS WEIGHTING
C     WITH THE ASSUMPTION THAT THE FREQUENCY RESPONSE DATA IS LOCATED
C     IN THE MIDDLE OF THE PLOT
C
C     NCUT- CUT-OFF NUMBER FOR THE LOW PASS FROM THE CENTER
C     NX  - X-COORDINATE IN THE MSIZE X MSIZE
C     NY  - Y-COORDINATE IN THE MSIZE X MSIZE
C     MSIZE- SIZE OF THE ARRAY
C
      FUNCTION WEIGHT(NCUT,NX,NY,MSIZE)
      CHARACTER LP
      PI=3.14159265
      X=ABS(FLOAT(NX-MSIZE/2-1))
      Y=ABS(FLOAT(NY-MSIZE/2-1))
      RADIUS=SQRT(X**2+Y**2)
      IF (RADIUS.LE.FLOAT(NCUT)) WEIGHT=0.5*(1.+COS(PI*RADIUS/FLOAT(NCUT)))
      IF (RADIUS.GT.FLOAT(NCUT)) WEIGHT=0.
      RETURN
      END
```

REFERENCES

[1] Shannon, C. E.: 'Communication in the presence of noise' IRE Proceedings, January 1949.

[2] Papoulis, A.: Fourier Integral and Its Applications McGraw-Hill, New York, 1962.

[3] Kennaugh and Cosgriff: 'The use of impulse response in electromagnetic scattering problems.' 1958 IRE National Convention Record Part 1, pp 72-77.

[4] Kennaugh and Moffatt: 'Transcient and impulse response approximation.' IEEE Proceedings, August 1965, pp 893-901.

[5] Lewis, R. M.: 'Physical optics inverse diffraction.' IEEE Transcations on Antenna and Propagation, Vol. AP-17, No. 3, May 1969, pp 308-314.

[6] Bojarski, N. N.: 'A survey of the physical optics inverse scattering identity.' IEEE Transcations on Antenna and Propagation, Vol. AP-30 No. 5, September 1982, pp 980-988.

[7] Petersen and Middleton: 'Sampling and reconstruction of wave-number limited functions in N-dimensional Euclidean Spaces.' Information and Control 5, 1962, pp 279-323.

[8] Mensa, Halevy and Wade: 'Coherent doppler tomography for microwave imaging.' IEEE Proceedings Vol. 71, No. 2, February 1983, pp 254-261.

[9] Coxeter, H. S. M.: 'Extreme forms.' Canadian Journal of Mathematics 3, 1951, pp 391-441.

[10] Harris, F. J.: 'On the use of window for harmonic analysis with the discrete Fourier transform.' IEEE Proceedings Vol. 66, No. 1, January 1978, pp 51-83.

[11] Dominek, A. K.: Personal communication.

[12] Walton, E. K.: Personal communication.

[13] Goodman, J. W.: Introduction to Fourier Optics
McGraw Hill, New York, 1968.

[14] Marhefka, R. J.: Personal communication.

[15] Peters, Leon: Personal communication.

[16] Kwan, B. W.: Personal communication.

END

FILMED

1-86

DTIC